Similarity Estimation Techniques from Rounding Algorithms

Moses S. Charikar
Dept. of Computer Science
Princeton University
35 Olden Street
Princeton, NJ 08544
moses@cs.princeton.edu

ABSTRACT

A locality sensitive hashing scheme is a distribution on a family $\mathcal{F}$ of hash functions operating on a collection of objects, such that for two objects x, y,

$$\mathbf{Pr}_{h \in \mathcal{F}}[h(x) = h(y)] = sim(x, y),$$

where $sim(x, y) \in [0, 1]$ is some similarity function defined on the collection of objects. Such a scheme leads to a compact representation of objects so that similarity of objects can be estimated from their compact sketches, and also leads to efficient algorithms for approximate nearest neighbor search and clustering. Min-wise independent permutations provide an elegant construction of such a locality sensitive hashing scheme for a collection of subsets with the set similarity measure $sim(A, B) = \frac{|A \cap B|}{|A \cup B|}$.

We show that rounding algorithms for LPs and SDPs used in the context of approximation algorithms can be viewed as locality sensitive hashing schemes for several interesting collections of objects. Based on this insight, we construct new locality sensitive hashing schemes for:

1. A collection of vectors with the distance between $\vec{u}$ and $\vec{v}$ measured by $\theta(\vec{u}, \vec{v})/\pi$, where $\theta(\vec{u}, \vec{v})$ is the angle between $\vec{u}$ and $\vec{v}$. This yields a sketching scheme for estimating the cosine similarity measure between two vectors, as well as a simple alternative to minwise independent permutations for estimating set similarity.
2. A collection of distributions on n points in a metric space, with distance between distributions measured by the Earth Mover Distance (**EMD**), (a popular distance measure in graphics and vision). Our hash functions map distributions to points in the metric space such that, for distributions P and Q,

$$\begin{aligned}\mathbf{EMD}(P, Q) \quad &\leq \quad \mathbf{E}_{h \in \mathcal{F}}[d(h(P), h(Q))] \\ &\leq O(\log n \log \log n) \cdot \mathbf{EMD}(P, Q).\end{aligned}$$

STOC'02, May 19-21, 2002, Montreal, Quebec, Canada.
Copyright 2002 ACM 1-58113-495-9/02/0005 ...$5.00.

1. INTRODUCTION

The current information explosion has resulted in an increasing number of applications that need to deal with large volumes of data. While traditional algorithm analysis assumes that the data fits in main memory, it is unreasonable to make such assumptions when dealing with massive data sets such as data from phone calls collected by phone companies, multimedia data, web page repositories and so on. This new setting has resulted in an increased interest in algorithms that process the input data in restricted ways, including sampling a few data points, making only a few passes over the data, and constructing a succinct sketch of the input which can then be efficiently processed.

There has been a lot of recent work on streaming algorithms, i.e. algorithms that produce an output by making one pass (or a few passes) over the data while using a limited amount of storage space and time. To cite a few examples, Alon *et al* [2] considered the problem of estimating frequency moments and Guha *et al* [25] considered the problem of clustering points in a streaming fashion. Many of these streaming algorithms need to represent important aspects of the data they have seen so far in a small amount of space; in other words they maintain a compact sketch of the data that encapsulates the relevant properties of the data set. Indeed, some of these techniques lead to sketching algorithms – algorithms that produce a compact sketch of a data set so that various measurements on the original data set can be estimated by efficient computations on the compact sketches. Building on the ideas of [2], Alon *et al* [1] give algorithms for estimating join sizes. Gibbons and Matias [18] give sketching algorithms producing so called *synopsis data structures* for various problems including maintaining approximate histograms, hot lists and so on. Gilbert *et al* [19] give algorithms to compute sketches for data streams so as to estimate any linear projection of the data and use this to get individual point and range estimates. Recently, Gilbert *et al* [21] gave efficient algorithms for the dynamic maintenance of histograms. Their algorithm processes a stream of updates and maintains a small sketch of the data from which the optimal histogram representation can be approximated very quickly.

In this work, we focus on sketching algorithms for estimating similarity, i.e. the construction of functions that produce succinct sketches of objects in a collection, such that the similarity of objects can be estimated efficiently from their sketches. Here, similarity $sim(x, y)$ is a function that maps pairs of objects x, y to a number in $[0,1]$, measuring the degree of similarity between x and y. $sim(x,y) = 1$ corresponds to objects x, y that are identical while $sim(x,y) = 0$ corresponds to objects that are very different.

Broder *et al* [8, 5, 7, 6] introduced the notion of *min-wise independent permutations*, a technique for constructing such sketching functions for a collection of sets. The similarity measure considered there was

$$sim(A,B) = \frac{|A \cap B|}{|A \cup B|}.$$

We note that this is exactly the Jaccard coefficient of similarity used in information retrieval.

The min-wise independent permutation scheme allows the construction of a distribution on hash functions $h : 2^U \to U$ such that

$$\mathbf{Pr}_{h \in \mathcal{F}}[h(A) = h(B)] = sim(A,B).$$

Here $\mathcal{F}$ denotes the family of hash functions (with an associated probability distribution) operating on subsets of the universe U. By choosing say t hash functions $h_1, \ldots h_t$ from this family, a set S could be represented by the hash vector $(h_1(S), \ldots h_t(S))$. Now, the similarity between two sets can be estimated by counting the number of matching coordinates in their corresponding hash vectors.[1]

The work of Broder *et al* was originally motivated by the application of eliminating near-duplicate documents in the Altavista index. Representing documents as sets of features with similarity between sets determined as above, the hashing technique provided a simple method for estimating similarity of documents, thus allowing the original documents to be discarded and reducing the input size significantly.

In fact, the minwise independent permutations hashing scheme is a particular instance of a *locality sensitive hashing scheme* introduced by Indyk and Motwani [31] in their work on nearest neighbor search in high dimensions.

Definition 1. *A locality sensitive hashing scheme is a distribution on a family $\mathcal{F}$ of hash functions operating on a collection of objects, such that for two objects x, y,*

$$\mathbf{Pr}_{h \in \mathcal{F}}[h(x) = h(y)] = sim(x,y) \quad (1)$$

Here $sim(x,y)$ is some similarity function defined on the collection of objects.

Given a hash function family $\mathcal{F}$ that satisfies (1), we will say that $\mathcal{F}$ is a locality sensitive hash function family corresponding to similarity function $sim(x,y)$. Indyk and Motwani showed that such a hashing scheme facilitates the construction of efficient data structures for answering approximate nearest-neighbor queries on the collection of objects.

In particular, using the hashing scheme given by minwise independent permutations results in efficient data structures for set similarity queries and leads to efficient clustering algorithms. This was exploited later in several experimental papers: Cohen *et al* [14] for association-rule mining, Haveliwala *et al* [27] for clustering web documents, Chen *et al* [13] for selectivity estimation of boolean queries, Chen *et al* [12] for twig queries, and Gionis *et al* [22] for indexing set value attributes. All of this work used the hashing technique for set similarity together with ideas from [31].

We note that the definition of *locality sensitive hashing* used by [31] is slightly different, although in the same spirit as our definition. Their definition involves parameters $r_1 > r_2$ and $p_1 > p_2$. A family $\mathcal{F}$ is said to be (r_1, r_2, p_1, p_2)-sensitive for a similarity measure $sim(x,y)$ if $\mathbf{Pr}_{h \in \mathcal{F}}[h(x) = h(y)] \geq p_1$ when $sim(x,y) \geq r_1$ and $\mathbf{Pr}_{h \in \mathcal{F}}[h(x) = h(y)] \leq p_2$ when $sim(x,y) \leq r_2$. Despite the difference in the precise definition, we chose to retain the name *locality sensitive hashing* in this work since the two notions are essentially the same. Hash functions with closely related properties were investigated earlier by Linial and Sasson [34] and Indyk *et al* [32].

1.1 Our Results

In this paper, we explore constructions of locality sensitive hash functions for various other interesting similarity functions. The utility of such hash function schemes (for nearest neighbor queries and clustering) crucially depends on the fact that the similarity estimation is based on a test of equality of the hash function values. We make an interesting connection between constructions of similarity preserving hash-functions and rounding procedures used in the design of approximation algorithms. We show that procedures used for rounding fractional solutions from linear programs and vector solutions to semidefinite programs can be used to derive similarity preserving hash functions for interesting classes of similarity functions.

In Section 2, we prove some necessary conditions on similarity measures $sim(x,y)$ for the existence of locality sensitive hash functions satisfying (1). Using this, we show that such locality sensitive hash functions do not exist for certain commonly used similarity measures in information retrieval, the Dice coefficient and the Overlap coefficient.

In seminal work, Goemans and Williamson [24] introduced semidefinite programming relaxations as a tool for approximation algorithms. They used the random hyperplane rounding technique to round vector solutions for the MAX-CUT problem. We will see in Section 3 that the random hyperplane technique naturally gives a family of hash functions $\mathcal{F}$ for vectors such that

$$\mathbf{Pr}_{h \in \mathcal{F}}[h(\vec{u}) = h(\vec{v})] = 1 - \frac{\theta(\vec{u}, \vec{v})}{\pi}.$$

Here $\theta(\vec{u}, \vec{v})$ refers to the angle between vectors $\vec{u}$ and $\vec{v}$. Note that the function $1 - \frac{\theta}{\pi}$ is closely related to the function $cos(\theta)$. (In fact it is always within a factor 0.878 from it. Moreover, $cos(\theta)$ can be estimated from an estimate of θ.) Thus this similarity function is very closely related to the cosine similarity measure, commonly used in information retrieval. (In fact, Indyk and Motwani [31] describe how the set similarity measure can be adapted to measure dot product between binary vectors in d-dimensional Hamming space. Their approach breaks up the data set into $O(\log d)$ groups, each consisting of approximately the same weight. Our approach, based on estimating the angle between vectors is more direct and is also more general since it applies to general vectors.) We also note that the cosine between vectors can be estimated from known techniques based on random projections [2, 1, 20]. However, the advantage of a locality sensitive hashing based scheme is that this directly yields techniques for nearest neighbor search for the cosine similarity measure.

[1] One question left open in [7] was the issue of compact representation of hash functions in this family; this was settled by Indyk [28], who gave a construction of a small family of minwise independent permutations.

An attractive feature of the hash functions obtained from the random hyperplane method is that the output is a single bit; thus the output of t hash functions can be concatenated very easily to produce a t-bit vector.[2] Estimating similarity between vectors amounts to measuring the Hamming distance between the corresponding t-bit hash vectors. We can represent sets by their characteristic vectors and use this locality sensitive hashing scheme for measuring similarity between sets. This yields a slightly different similarity measure for sets, one that is linearly proportional to the angle between their characteristic vectors.

In Section 4, we present a locality sensitive hashing scheme for a certain metric on distributions on points, called the *Earth Mover Distance*. We are given a set of points $L = \{l_1, \ldots l_n\}$, with a distance function $d(i,j)$ defined on them. A probability distribution $P(X)$ (or distribution for short) is a set of weights $p_1, \ldots p_n$ on the points such that $p_i \geq 0$ and $\sum p_i = 1$. (We will often refer to distribution $P(X)$ as simply P, implicitly referring to an underlying set X of points.) The *Earth Mover Distance* $\mathbf{EMD}(P,Q)$ between two distributions P and Q is defined to be the cost of the min cost matching that transforms one distribution to another. (Imagine each distribution as placing a certain amount of earth on each point. $\mathbf{EMD}(P,Q)$ measures the minimum amount of work that must be done in transforming one distribution to the other.) This is a popular metric for images and is used for image similarity, navigating image databases and so on [37, 38, 39, 40, 36, 15, 16, 41, 42]. The idea is to represent an image as a distribution on features with an underlying distance metric on features (e.g. colors in a color spectrum). Since the earth mover distance is expensive to compute (requiring a solution to a minimum transportation problem), applications typically use an approximation of the earth mover distance. (e.g. representing distributions by their centroids).

We construct a hash function family for estimating the earth mover distance. Our family is based on rounding algorithms for LP relaxations for the problem of classification with pairwise relationships studied by Kleinberg and Tardos [33], and further studied by Calinescu *et al* [10] and Chekuri *et al* [11]. Combining a new LP formulation described by Chekuri *et al* together with a rounding technique of Kleinberg and Tardos, we show a construction of a hash function family which approximates the earth mover distance to a factor of $O(\log n \log \log n)$. Each hash function in this family maps a distribution on points $L = \{l_1, \ldots, l_n\}$ to some point l_i in the set. For two distributions $P(X)$ and $Q(X)$ on the set of points, our family of hash functions $\mathcal{F}$ satisfies the property that:

$$\begin{aligned}\mathbf{EMD}(P,Q) \quad &\leq \quad \mathbf{E}_{h\in\mathcal{F}}[d(h(P),h(Q))] \\ &\leq O(\log n \log\log n)\cdot \mathbf{EMD}(P,Q).\end{aligned}$$

We also show an interesting fact about a rounding algorithm in Kleinberg and Tardos [33] applying to the case where the underlying metric on points is a uniform metric. In this case, we show that their rounding algorithm can be viewed as a generalization of min-wise independent permutations extended to a continuous setting. Their rounding procedure yields a locality sensitive hash function for vectors whose coordinates are all non-negative. Given two vectors $\vec{a} = (a_1, \ldots a_n)$ and $\vec{b} = (b_1, \ldots b_n)$, the similarity function is

$$sim(\vec{a},\vec{b}) = \frac{\sum_i \min(a_i,b_i)}{\sum_i \max(a_i,b_i)}.$$

(Note that when $\vec{a}$ and $\vec{b}$ are the characteristic vectors for sets A and B, this expression reduces to the set similarity measure for min-wise independent permutations.)

Applications of locality sensitive hash functions to solving nearest neighbor queries typically reduce the problem to the Hamming space. Indyk and Motwani [31] give a data structure that solves the approximate nearest neighbor problem on the Hamming space. Their construction is a reduction to the so called PLEB (Point Location in Equal Balls) problem, followed by a hashing technique concatenating the values of several locality sensitive hash functions. We give a simple technique that achieves the same performance as the Indyk Motwani result in Section 5. The basic idea is as follows: Given bit vectors consisting of d bits each, we choose a number of random permutations of the bits. For each random permutation σ, we maintain a sorted order of the bit vectors, in lexicographic order of the bits permuted by σ. To find a nearest neighbor for a query bit vector q we do the following: For each permutation σ, we perform a binary search on the sorted order corresponding to σ to locate the bit vectors closest to q (in the lexicographic order obtained by bits permuted by σ). Further, we search in each of the sorted orders proceeding upwards and downwards from the location of q, according to a certain rule. Of all the bit vectors examined, we return the one that has the smallest Hamming distance to the query vector. The performance bounds we can prove for this simple scheme are identical to that proved by Indyk and Motwani for their scheme.

2. EXISTENCE OF LOCALITY SENSITIVE HASH FUNCTIONS

In this section, we discuss certain necessary properties for the existence of locality sensitive hash function families for given similarity measures.

Lemma 1. *For any similarity function $sim(x,y)$ that admits a locality sensitive hash function family as defined in (1), the distance function $1 - sim(x,y)$ satisfies triangle inequality.*

Proof. Suppose there exists a locality sensitive hash function family such that

$$\mathbf{Pr}_{h\in\mathcal{F}}[h(x) = h(y)] = sim(x,y).$$

Then,

$$1 - sim(x,y) = \mathbf{Pr}_{h\in\mathcal{F}}[h(x) \neq h(y)].$$

Let $\Delta_h(x,y)$ be an indicator variable for the event $h(x) \neq h(y)$. We claim that $\Delta_h(x,y)$ satisfies the triangle inequality, i.e.

$$\Delta_h(x,y) + \Delta_h(y,z) \geq \Delta_h(x,z).$$

[2] In Section 2, we will show that we can convert any locality sensitive hashing scheme to one that maps objects to $\{0,1\}$ with a slight change in similarity measure. However, the modified hash functions convey less information, e.g. the collision probability for the modified hash function family is at least $1/2$ even for a pair of objects with original similarity 0.

Since $\Delta_h()$ takes values in the set $\{0, 1\}$, the only case when the above inequality could be violated would be when $\Delta_h(x, y) = \Delta_h(y, z) = 0$. But in this case $h(x) = h(y)$ and $h(y) = h(z)$. Thus, $h(x) = h(z)$ implying that $\Delta_h(x, z) = 0$ and the inequality is satisfied. This proves the claim. Now,

$$1 - sim(x, y) = \mathbf{E}_{h\in\mathcal{F}}[\Delta_h(x, y)]$$

Since $\Delta_h(x, y)$ satisfies the triangle inequality, $\mathbf{E}_{h\in\mathcal{F}}[\Delta_h(x, y)]$ must also satisfy the triangle inequality. This proves the lemma. □

This gives a very simple proof of the fact that for the set similarity measure $sim(A, B) = \frac{|A\cap B|}{|A\cup B|}$, $1 - sim(A, B)$ satisfies the triangle inequality. This follows from Lemma 1 and the fact that a set similarity measure admits a locality sensitive hash function family, namely that given by minwise independent permutations.

One could ask the question whether locality sensitive hash functions satisfying the definition (1) exist for other commonly used set similarity measures in information retrieval. For example, Dice's coefficient is defined as

$$sim_{\text{D}ice}(A, B) = \frac{|A \cap B|}{\frac{1}{2}(|A| + |B|)}$$

The Overlap coefficient is defined as

$$sim_{\text{O}vl}(A, B) = \frac{|A \cap B|}{\min(|A|, |B|)}$$

We can use Lemma 1 to show that there is no such locality sensitive hash function family for Dice's coefficient and the Overlap measure by showing that the corresponding distance function does not satisfy triangle inequality.

Consider the sets $A = \{a\}, B = \{b\}, C = \{a, b\}$. Then,

$$\begin{aligned} sim_{\text{D}ice}(A, C) = \frac{2}{3}, \qquad & sim_{\text{D}ice}(C, B) = \frac{2}{3}, \\ sim_{\text{D}ice}(A, B) &= 0 \\ 1 - sim_{\text{D}ice}(A, C) &+ 1 - sim_{\text{D}ice}(C, B) \\ &< 1 - sim_{\text{D}ice}(A, B) \end{aligned}$$

Similarly, the values for the Overlap measure are as follows:

$$sim_{\text{O}vl}(A, C) = 1, \quad sim_{\text{O}vl}(C, B) = 1, \quad sim_{\text{O}vl}(A, B) = 0$$
$$1 - sim_{\text{O}vl}(A, C) + 1 - sim_{\text{O}vl}(C, B) < 1 - sim_{\text{O}vl}(A, B)$$

This shows that there is no locality sensitive hash function family corresponding to Dice's coefficient and the Overlap measure.

It is often convenient to have a hash function family that maps objects to $\{0, 1\}$. In that case, the output of t different hash functions can simply be concatenated to obtain a t-bit hash value for an object. In fact, we can always obtain such a binary hash function family with a slight change in the similarity measure. A similar result was used and proved by Gionis *et al* [22]. We include a proof for completeness.

LEMMA 2. *Given a locality sensitive hash function family $\mathcal{F}$ corresponding to a similarity function $sim(x, y)$, we can obtain a locality sensitive hash function family $\mathcal{F}'$ that maps objects to $\{0, 1\}$ and corresponds to the similarity function $\frac{1+sim(x,y)}{2}$.*

PROOF. Suppose we have a hash function family such that

$$\mathbf{Pr}_{h\in\mathcal{F}}[h(x) = h(y)] = sim(x, y).$$

Let $\mathcal{B}$ be a pairwise independent family of hash functions that operate on the domain of the functions in $\mathcal{F}$ and map elements in the domain to $\{0, 1\}$. Then $\mathbf{Pr}_{b\in\mathcal{B}}[b(u) = b(v)] = 1/2$ if $u \neq v$ and $\mathbf{Pr}_{b\in\mathcal{B}}[b(u) = b(v)] = 1$ if $u = v$. Consider the hash function family obtained by composing a hash function from $\mathcal{F}$ with one from $\mathcal{B}$. This maps objects to $\{0, 1\}$ and we claim that it has the required properties.

$$\mathbf{Pr}_{h\in\mathcal{F}, b\in\mathcal{B}}[b(h(x)) = b(h(y))] = \frac{1 + sim(x, y)}{2}$$

With probability $sim(x, y)$, $h(x) = h(y)$ and hence $b(h(x) = b(h(y))$. With probability $1 - sim(x, y)$, $h(x) \neq h(y)$ and in this case, $\mathbf{Pr}_{b\in\mathcal{B}}[b(h(x) = b(h(y))] = \frac{1}{2}$. Thus,

$$\begin{aligned} \mathbf{Pr}[b(h(x)) = b(h(y))] &= sim(x, y) + (1 - sim(x, y))/2 \\ &= (1 + sim(x, y))/2. \end{aligned}$$

□

This can be used to show a stronger condition for the existence of a locality sensitive hash function family.

LEMMA 3. *For any similarity function $sim(x, y)$ that admits a locality sensitive hash function family as defined in (1), the distance function $1 - sim(x, y)$ is isometrically embeddable in the Hamming cube.*

PROOF. Firstly, we apply Lemma 2 to construct a binary locality sensitive hash function family corresponding to similarity function $sim'(x, y) = (1 + sim(x, y))/2$. Note that such a binary hash function family gives an embedding of objects into the Hamming cube (obtained by concatenating the values of all the hash functions in the family). For object x, let $v(x)$ be the element in the Hamming cube x is mapped to. $1 - sim'(x, y)$ is simply the fraction of bits that do not agree in $v(x)$ and $v(y)$, which is proportional to the Hamming distance between $v(x)$ and $v(y)$. Thus this embedding is an isometric embedding of the distance function $1 - sim'(x, y)$ in the Hamming cube. But

$$1 - sim'(x, y) = 1 - (1 + sim(x, y))/2 = (1 - sim(x, y))/2.$$

This implies that $1 - sim(x, y)$ can be isometrically embedded in the Hamming cube. □

We note that Lemma 3 has a weak converse, i.e. for a similarity measure $sim(x, y)$ any isometric embedding of the distance function $1 - sim(x, y)$ in the Hamming cube yields a locality sensitive hash function family corresponding to the similarity measure $(\alpha + sim(x, y))/(\alpha + 1)$ for some $\alpha > 0$.

3. RANDOM HYPERPLANE BASED HASH FUNCTIONS FOR VECTORS

Given a collection of vectors in R^d, we consider the family of hash functions defined as follows: We choose a random vector $\vec{r}$ from the d-dimensional Gaussian distribution (i.e. each coordinate is drawn the 1-dimensional Gaussian distribution). Corresponding to this vector $\vec{r}$, we define a hash function $h_{\vec{r}}$ as follows:

$$h_{\vec{r}}(\vec{u}) = \begin{cases} 1 & \text{if } \vec{r}\cdot\vec{u} \geq 0 \\ 0 & \text{if } \vec{r}\cdot\vec{u} < 0 \end{cases}$$

Then for vectors $\vec{u}$ and $\vec{v}$,

$$\mathbf{Pr}[h_{\vec{r}}(\vec{u}) = h_{\vec{r}}(\vec{v})] = 1 - \frac{\theta(\vec{u},\vec{v})}{\pi}.$$

This was used by Goemans and Williamson [24] in their rounding scheme for the semidefinite programming relaxation of MAX-CUT.

Picking a random hyperplane amounts to choosing a normally distributed random variable for each dimension. Thus even representing a hash function in this family could require a large number of random bits. However, for n vectors, the hash functions can be chosen by picking $O(\log^2 n)$ random bits, i.e. we can restrict the random hyperplanes to be in a family of size $2^{O(\log^2 n)}$. This follows from the techniques in Indyk [30] and Engebretsen *et al* [17], which in turn use Nisan's pseudorandom number generator for space bounded computations [35]. We omit the details since they are similar to those in [30, 17].

Using this random hyperplane based hash function, we obtain a hash function family for set similarity, for a slightly different measure of similarity of sets. Suppose sets are represented by their characteristic vectors. Then, applying the above scheme gives a locality sensitive hashing scheme where

$$\begin{aligned} \mathbf{Pr}[h(A) = h(B)] &= 1 - \frac{\theta}{\pi}, \text{ where} \\ \theta &= \cos^{-1}\left(\frac{|A \cap B|}{\sqrt{|A|\cdot|B|}}\right) \end{aligned}$$

Also, this hash function family facilitates easy incorporation of element weights in the similarity calculation, since the values of the coordinates of the characteristic vectors could be real valued element weights. Later, in Section 4.1 we will present another technique to define and estimate similarity of weighted sets.

4. THE EARTH MOVER DISTANCE

Consider a set of points $L = \{l_1, \ldots l_n\}$ with a distance function $d(i,j)$ (assumed to be a metric). A distribution $P(L)$ on L is a collection of non-negative weights $(p_1, \ldots p_n)$ for points in X such that $\sum p_i = 1$. The distance between two distributions $P(L)$ and $Q(L)$ is defined to be the optimal cost of the following minimum transportation problem:

$$\min \sum_{i,j} \mathbf{f}_{i,j} \cdot d(i,j) \tag{2}$$

$$\forall\, i \quad \sum_j \mathbf{f}_{i,j} = p_i \tag{3}$$

$$\forall\, j \quad \sum_i \mathbf{f}_{i,j} = q_j \tag{4}$$

$$\forall\, i,j \quad \mathbf{f}_{i,j} \geq 0 \tag{5}$$

Note that we define a somewhat restricted form of the Earth Mover Distance. The general definition does not assume that the sum of the weights is identical for distributions $P(L)$ and $Q(L)$. This is useful for example in matching a small image to a portion of a larger image.

We will construct a hash function family for estimating the Earth Mover Distance based on rounding algorithms for the problem of classification with pairwise relationships, introduced by Kleinberg and Tardos [33]. (A closely related problem was also studied by Broder *et al* [9]). In designing hash functions to estimate the Earth Mover Distance, we will relax the definition of locality sensitive hashing (1) in three ways.

1. Firstly, the quantity we are trying to estimate is a distance measure, not a similarity measure in $[0,1]$.

2. Secondly, we will allow the hash functions to map objects to points in a metric space and measure $\mathbf{E}[d(h(x), h(y))]$. (A locality sensitive hash function for a similarity measure $sim(x,y)$ can be viewed as a scheme to estimate the distance $1 - sim(x,y)$ by $\mathbf{Pr}_{h\in\mathcal{F}}[h(x) \neq h(y)]$. This is equivalent to having a uniform metric on the hash values).

3. Thirdly, our estimator for the Earth Mover Distance will not be an unbiased estimator, i.e. our estimate will approximate the Earth Mover Distance to within a small factor.

We now describe the problem of classification with pairwise relationships. Given a collection of objects V and labels $L = \{l_1, \ldots, l_n\}$, the goal is to assign labels to objects. The cost of assigning label l to object $u \in V$ is $c(u,l)$. Certain pairs of objects (u,v) are related; such pairs form the edges of a graph over V. Each edge $e = (u,v)$ is associated with a non-negative weight w_e. For edge $e = (u,v)$, if u is assigned label $h(u)$ and v is assigned label $h(v)$, then the cost paid is $w_e d(h(u), h(v))$.

The problem is to come up with an assignment of labels $h : V \to L$, so as to minimize the cost of the labeling h given by

$$\sum_{u\in V} c(v, h(v)) + \sum_{e=(u,v)\in E} w_e d(h(u), h(v))$$

The approximation algorithms for this problem use an LP to assign, for every $u \in V$, a probability distribution over labels in L (i.e. a set of non-negative weights that sum up to 1). Given a distribution P over labels in L, the rounding algorithm of Kleinberg and Tardos gave a randomized procedure for assigning label $h(P)$ to P with the following properties:

1. Given distribution $P(L) = (p_1, \ldots p_n)$,
$$\mathbf{Pr}[h(P) = l_i] = p_i. \tag{6}$$

2. Suppose P and Q are probability distributions over L.
$$\mathbf{E}[d(h(P), h(Q))] \leq O(\log n \log\log n)\, \mathbf{EMD}(P,Q) \tag{7}$$

We note that the second property (7) is not immediately obvious from [33], since they do not describe LP relaxations for general metrics. Their LP relaxations are defined for Hierarchically well Separated Trees (HSTs). They convert a general metric to such an HST using Bartal's results [3, 4] on probabilistic approximation of metric spaces via tree metrics. However, it follows from combining ideas in [33] with those in Chekuri *et al* [11]. Chekuri *et al* do in fact give an LP relaxation for general metrics. The LP relaxation does indeed produce distributions over labels for every object $u \in V$. The *fractional distance* between two labelings is expressed as the min cost transshipment between P and Q, which is identical to the Earth Mover Distance $\mathbf{EMD}(P,Q)$. Now, this fractional solution can be used in the rounding algorithm developed by Kleinberg and Tardos to obtain the second property (7) claimed above. In fact, Chekuri *et al* use this fact to claim that the gap of their LP relaxation is at most $O(\log n \log\log n)$ (Theorem 5.1 in [11]).

We elaborate some more on why the property (7) holds. Kleinberg and Tardos first (probabilistically) approximate the metric on L by an HST using [3, 4]. This is a tree with all vertices in the original metric at the leaves. The pairwise distance between any two vertices does no decrease and all pairwise distances are increased by a factor of at most $O(\log n \log\log n)$ (in expectation). For this tree metric, they use an LP formulation which can be described as follows. Suppose we have a rooted tree. For subtree T, let ℓ_T denote the length of the edge that T hangs off of, i.e. the first edge on the path from T to the root. Further, for distribution P on the vertices of the original metric, let $P(T)$ denote the total probability mass that P assigns to leaves in T; $Q(T)$ is similarly defined. The distance between distributions P and Q is measured by $\sum_T \ell_T |P(T) - Q(T)|$, where the summation is computed over all subtrees T. The Kleinberg Tardos rounding scheme ensures that $\mathbf{E}[d(h(P),h(Q))]$ is within a constant factor of $\sum_T \ell_T |P(T) - Q(T)|$.

Suppose instead, we measured the distance between distributions by $\mathbf{EMD}(P,Q)$, defined on the original metric. By probabilistically approximating the original metric by a tree metric T', the expected value of the distance $\mathbf{EMD}_{T'}(P,Q)$ (on the tree metric T') is at most a factor of $O(\log n \log\log n)$ times $\mathbf{EMD}(P,Q)$. This follows since all distances increase by $O(\log n \log\log n)$ in expectation. Now note that the tree distance measure used by Kleinberg and Tardos $\sum_T \ell_T |P(T) - Q(T)|$ is a lower bound on (and in fact exactly equal to) $\mathbf{EMD}_{T'}(P,Q)$. To see that this is a lower bound, note that in the min cost transportation between P and Q on T', the flow on the edge leading upwards from subtree T must be at least $|P(T) - Q(T)|$. Since the rounding scheme ensures that $\mathbf{E}[d(h(P),h(Q))]$ is within a constant factor of $\sum_T \ell_T |P(T) - Q(T)|$, we have that

$$\begin{aligned}\mathbf{E}[d(h(P),h(Q))] &\leq O(1)\,\mathbf{EMD}_{T'}(P,Q)\\ &\leq O(\log n \log\log n)\,\mathbf{EMD}(P,Q)\end{aligned}$$

where the expectation is over the random choice of the HST and the random choices made by the rounding procedure.

THEOREM 1. *The Kleinberg Tardos rounding scheme yields a locality sensitive hashing scheme such that*

$$\begin{aligned}\mathbf{EMD}(P,Q) \;\leq\; & \mathbf{E}[d(h(P),h(Q))]\\ & \leq O(\log n \log\log n)\,\mathbf{EMD}(P,Q).\end{aligned}$$

PROOF. The upper bound on $\mathbf{E}[d(h(P),h(Q))]$ follows directly from the second property (7) of the rounding scheme stated above.

We show that the lower bound follows from the first property (6). Let $y_{i,j}$ be the joint probability that $h(P) = l_i$ and $h(Q) = l_j$. Note that $\sum_j y_{i,j} = p_i$, since this is simply the probability that $h(P) = l_i$. Similarly $\sum_i y_{i,j} = q_j$, since this is simply the probability that $h(Q) = l_j$. Now, if $h(P) = l_i$ and $h(Q) = l_j$, then $d(h(P)h(Q)) = d(i,j)$. Hence $\mathbf{E}[d(f(P),f(Q))] = \sum_{i,j} y_{i,j} \cdot d(i,j)$. Let us write down the expected cost and the constraints on $y_{i,j}$.

$$\begin{aligned}\mathbf{E}[d(h(P),h(Q))] &= \sum_{i,j} \mathbf{y}_{i,j} \cdot d(i,j)\\ \forall\, i \quad \sum_j y_{i,j} &= p_i\\ \forall\, j \quad \sum_i y_{i,j} &= q_j\\ \forall\, i,j \quad y_{i,j} &\geq 0\end{aligned}$$

Comparing this with the LP for $\mathbf{EMD}(P,Q)$, we see that the values of $f_{i,j} = y_{i,j}$ is a feasible solution to the LP (2) to (5) and $\mathbf{E}[d(h(P),h(Q))]$ is exactly the value of this solution. Since $\mathbf{EMD}(P,Q)$ is the minimum value of a feasible solution, it follows that $\mathbf{EMD}(P,Q) \leq \mathbf{E}[d(h(P),h(Q))]$. □

Calinescu *et al* [10] study a variant of the classification problem with pairwise relationships called the 0-extension problem. This is the version without assignment costs where some objects are assigned labels apriori and this labeling must be extended to the other objects (a generalization of multiway cut). For this problem, they design a rounding scheme to get a $O(\log n)$ approximation. Again, their technique does not explicitly use an LP that gives probability distributions on labels. However in hindsight, their rounding scheme can be interpreted as a randomized procedure for assigning labels to distributions such that

$$\mathbf{E}[d(h(P),h(Q))] \leq O(\log n)\,\mathbf{EMD}(P,Q).$$

Thus their rounding scheme gives a tighter guarantee than (7). However, they do not ensure (6). Thus the previous proof showing that $\mathbf{EMD}(P,Q) \leq \mathbf{E}[d(h(P),h(Q))]$ does not apply. In fact one can construct examples such that $\mathbf{EMD}(P,Q) > 0$, yet $\mathbf{E}[d(h(P),h(Q))] = 0$. Hence, the resulting hash function family provides an upper bound on $\mathbf{EMD}(P,Q)$ within a factor $O(\log n)$ but does not provide a good lower bound.

We mention that the hashing scheme described provides an approximation to the Earth Mover Distance where the quality of the approximation is exactly the factor by which the underlying metric can be probabilistically approximated by HSTs. In particular, if the underlying metric itself is an HST, this yields an estimate within a constant factor. This could have applications in compactly representing distributions over hierarchical classes. For example, documents can be assigned a probability distribution over classes in the Open Directory Project (ODP) hierarchy. This hierarchy could be thought of as an HST and documents can be mapped to distributions over this HST. The distance between two distributions can be measured by the Earth Mover Distance. In this case, the hashing scheme described gives a way to estimate this distance measure to within a constant factor.

4.1 Weighted Sets

We show that the Kleinberg Tardos [33] rounding scheme for the case of the uniform metric actually is an extension of min-wise independent permutations to the weighted case.

First we recall the hashing scheme given by min-wise independent permutations. Given a universe U, consider a random permutation π of U. Assume that the elements of U are totally ordered. Given a subset $A \subseteq U$, the hash function h_π is defined as follows:

$$h_\pi(A) = \min\{\pi(A)\}$$

Then the property satisfied by this hash function family is that

$$\mathbf{Pr}_\pi[h_\pi(A) = h_\pi(B)] = \frac{|A \cap B|}{|A \cup B|}$$

We now review the Kleinberg Tardos rounding scheme for the uniform metric: Firstly, imagine that we pick an infinite sequence $\{(i_t, \alpha_t)\}_{t=1}^{\infty}$ where for each t, i_t is picked uniformly and at random in $\{1, \ldots n\}$ and α_t is picked uniformly and at random in $[0, 1]$. Given a distribution $P = (p_1, \ldots, p_n)$, the assignment of labels is done in phases. In the ith phase, we check whether $\alpha_i \leq p_{i_t}$. If this is the case and P has not been assigned a label yet, it is assigned label i_t.

Now, we can think of these distributions as sets in R^2 (see Figure 1).

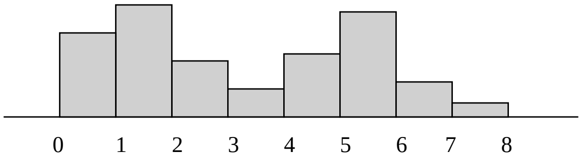

Figure 1: *Viewing a distribution as a continuous set.*

The set $S(P)$ corresponding to distribution P consists of the union of the rectangles $[i-1, i] \times [0, p_i]$. The elements of the universe are $[i-1, i] \times \alpha$. $[i-1, i] \times \alpha$ belongs to $S(P)$ iff $\alpha \leq p_i$. The notion of *cardinality* of union and intersection of sets is replaced by the *area* of the intersection and union of two such sets in R^2. Note that the Kleinberg Tardos rounding scheme can be interpreted as constructing a permutation of the universe and assigning to a distribution P, the value i such that (i, α) is the minimum in the permutation amongst all elements contained in $S(P)$. Suppose instead, we assign to P, the element (i, α) which is the minimum in the permutation of $S(P)$. Let h be a hash function derived from this scheme (a slight modification of the one in [33]). Then,

$$\mathbf{Pr}[h(P) = h(Q)] = \frac{|S(P) \cap S(Q)|}{|S(P) \cup S(Q)|} = \frac{\sum_i \min(p_i, q_i)}{\sum_i \max(p_i, q_i)} \quad (8)$$

For the Kleinberg Tardos rounding scheme, the probability of collision is at least the probability of collision for the modified scheme (since two objects hashed to (i, α_1) and (i, α_2) respectively in the modified scheme would be both mapped to i in the original scheme). Hence

$$\begin{aligned}
\mathbf{Pr}_{KT}[h(P) = h(Q)] &\geq \frac{\sum_i \min(p_i, q_i)}{\sum_i \max(p_i, q_i)} \\
\mathbf{Pr}_{KT}[h(P) \neq h(Q)] &\leq 1 - \frac{\sum_i \min(p_i, q_i)}{\sum_i \max(p_i, q_i)} \\
&= \frac{\sum_i |p_i - q_i|}{\sum_i \max(p_i, q_i)} \leq \sum_i |p_i - q_i|
\end{aligned}$$

The last inequality follows from the fact that $\sum p_i = \sum q_i = 1$ in the Kleinberg Tardos setting. This was exactly the property used in [33] to obtain a 2-approximation for the uniform metric case.

Note that the hashing scheme given by (8) is a generalization of min-wise independent permutations to the weighted setting where elements in sets are associated with weights $\in [0, 1]$. Min-wise independent permutations are a special case of this scheme when the weights are $\{0, 1\}$. This scheme could be useful in a setting where a weighted set similarity notion is desired. We note that the original min-wise independent permutations can be used in the setting of integer weights by simply duplicating elements according to their weight. The present scheme would work for any non-negative real weights.

5. APPROXIMATE NEAREST NEIGHBOR SEARCH IN HAMMING SPACE.

Applications of locality sensitive hash functions to solving nearest neighbor queries typically reduce the problem to the Hamming space. Indyk and Motwani [31] give a data structure that solves the approximate nearest neighbor problem on the Hamming space H^d. Their construction is a reduction to the so called PLEB (Point Location in Equal Balls) problem, followed by a hashing technique concatenating the values of several locality sensitive hash functions.

Theorem 2 ([31]). *For any $\epsilon > 0$, there exists an algorithm for ϵ-PLEB in H^d using $O(dn + n^{1+1/(1+\epsilon)})$ space and $O(n^{1/(1+\epsilon)})$ hash function evaluations for each query.*

We give a simple technique that achieves the same performance as the Indyk Motwani result:

Given bit vectors consisting of d bits each, we choose $N = O(n^{1/(1+\epsilon)})$ random permutations of the bits. For each random permutation σ, we maintain a sorted order O_σ of the bit vectors, in lexicographic order of the bits permuted by σ. Given a query bit vector q, we find the approximate nearest neighbor by doing the following: For each permutation σ, we perform a binary search on O_σ to locate the two bit vectors closest to q (in the lexicographic order obtained by bits permuted by σ). We now search in each of the sorted orders O_σ examining elements above and below the position returned by the binary search in order of the length of the longest prefix that matches q. This can be done by maintaining two pointers for each sorted order O_σ (one moves up and the other down). At each step we move one of the pointers up or down corresponding to the element with the longest matching prefix. (Here the length of the longest matching prefix in O_σ is computed relative to q with its bits permuted by σ). We examine $2N = O(n^{1/(1+\epsilon)})$ bit vectors in this way. Of all the bit vectors examined, we return the one that has the smallest Hamming distance to q. The performance bounds we can prove for this simple scheme are identical to that proved by Indyk and Motwani for their scheme. An advantage of this scheme is that we do not need a reduction to many instances of PLEB for different values of radius r, i.e. we solve the nearest neighbor problem simultaneously for all values of radius r using a single data structure.

We outline the main ideas of the analysis. In fact, the proof follows along similar lines to the proofs of Theorem 5 and Corollary 3 in [31]. Suppose the nearest neighbor of q is at a Hamming distance of r from q. Set $p_1 = 1 - \frac{r}{d}$, $p_2 = 1 - \frac{r(1+\epsilon)}{d}$ and $k = \log_{1/p_2} n$. Let $\rho = \frac{\ln 1/p_1}{\ln 1/p_2}$. Then $n^\rho = O(n^{1/(1+\epsilon)})$. We can show that with constant probability, from amongst $N = O(n^{1/(1+\epsilon)})$ permutations, there exists a permutation such that the nearest neighbor agrees with p on the first k coordinates in σ. Further, over all L permutations, the number of bit vectors that are at Hamming distance of more than $r(1+\epsilon)$ from q and agree on the first k coordinates is at most $2N$ with constant probability. This implies that for this permutation σ, one of the $2L$ bit vectors near q in the ordering O_σ and examined by the algorithm will be a $(1+\epsilon)$-approximate nearest neighbor. The probability calculations are similar to those in [31], and we only sketch the main ideas.

For any point q' at distance at least $r(1+\epsilon)$ from q, the probability that a random coordinate agrees with q is at most p_2. Thus the probability that the first k coordinates agree is at most $p_2^k = \frac{1}{n}$. For the N permutations, the expected number of such points that agree in the first k coordinates is at most N. The probability that this number is $\leq 2N$ is $> 1/2$. Further, for a random permutation σ, the probability that the nearest neighbor agrees in k coordinates is $p_1^k = n^{-\rho}$. Hence the probability that there exists one permutation amongst the $N = n^\rho$ permutations where the nearest neighbor agrees in k coordinates is at least $1 - (1 - n^{-\rho})^{n^\rho} > 1/2$. This establishes the correctness of the procedure.

As we stated earlier, a nice property of this data structure is that it automatically adjusts to the correct distance r to the nearest neighbor, i.e. we do not need to maintain separate data structures for different values of r.

6. CONCLUSIONS

We have demonstrated an interesting relationship between rounding algorithms used for rounding fractional solutions of LPs and vector solutions of SDPs on the one hand, and the construction of locality sensitive hash functions for interesting classes of objects, on the other.

Rounding algorithms yield new constructions of locality sensitive hash functions that were not known previously. Conversely (at least in hindsight), locality sensitive hash functions lead to rounding algorithms (as in the case of minwise independent permutations and the uniform metric case in Kleinberg and Tardos [33]).

An interesting direction to pursue would be to investigate the construction of sketching functions that allow one to estimate information theoretic measures of distance between distributions such as the KL-divergence, commonly used in statistical learning theory. Since the KL-divergence is neither symmetric nor satisfies triangle inequality, new ideas would be required in order to design a sketch function to approximate it. Such a sketch function, if one exists, would be a very valuable tool in compactly representing complex distributions.

7. REFERENCES

[1] N. Alon, P. B. Gibbons, Y. Matias, and M. Szegedy. Tracking Join and Self-Join Sizes in Limited Storage. *Proc. 18th PODS* pp. 10-20, 1999.

[2] N. Alon, Y. Matias, and M. Szegedy. The Space Complexity of Approximating the Frequency Moments. *JCSS* 58(1): 137-147, 1999

[3] Y. Bartal. Probabilistic approximation of metric spaces and its algorithmic application. *Proc. 37th FOCS*, pages 184–193, 1996.

[4] Y. Bartal. On approximating arbitrary metrics by tree metrics. In Proc. 30th STOC, pages 161–168, 1998.

[5] A. Z. Broder. On the resemblance and containment of documents. *Proc. Compression and Complexity of SEQUENCES*, pp. 21–29. IEEE Computer Society, 1997.

[6] A. Z. Broder. Filtering near-duplicate documents. *Proc. FUN 98*, 1998.

[7] A. Z. Broder, M. Charikar, A. Frieze, and M. Mitzenmacher. Min-wise independent permutations. *Proc. 30th STOC*, pp. 327–336, 1998.

[8] A. Z. Broder, S. C. Glassman, M. S. Manasse, and G. Zweig. Syntactic clustering of the Web. *Proc. 6th Int'l World Wide Web Conference*, pp. 391–404, 1997.

[9] A. Z. Broder, R. Krauthgamer, and M. Mitzenmacher. Improved classification via connectivity information. *Proc. 11th SODA*, pp. 576-585, 2000.

[10] G. Calinescu, H. J. Karloff, and Y. Rabani. Approximation algorithms for the 0-extension problem. *Proc. 11th SODA*, pp. 8-16, 2000.

[11] C. Chekuri, S. Khanna, J. Naor, and L. Zosin. Approximation algorithms for the metric labeling problem via a new linear programming formulation. *Proc. 12th SODA*, pp. 109-118, 2001.

[12] Z. Chen, H. V. Jagadish, F. Korn, N. Koudas, S. Muthukrishnan, R. T. Ng, and D. Srivastava. Counting Twig Matches in a Tree. *Proc. 17th ICDE* pp. 595-604. 2001.

[13] Z. Chen, F. Korn, N. Koudas, and S. Muthukrishnan. Selectivity Estimation for Boolean Queries. *Proc. 19th PODS*, pp. 216-225, 2000.

[14] E. Cohen, M. Datar, S. Fujiwara, A. Gionis, P. Indyk, R. Motwani, J. D. Ullman, and C. Yang. Finding Interesting Associations without Support Pruning. *Proc. 16th ICDE* pp. 489-499, 2000.

[15] S. Cohen and L. Guibas. The Earth Mover's Distance under Transformation Sets. *Proc. 7th IEEE Intnl. Conf. Computer Vision*, 1999.

[16] S. Cohen and L. Guibas. The Earth Mover's Distance: Lower Bounds and Invariance under Translation. Tech. report STAN-CS-TR-97-1597, Dept. of Computer Science, Stanford University, 1997.

[17] L. Engebretsen, P. Indyk and R. O'Donnell. Derandomized dimensionality reduction with applications. To appear in *Proc. 13th SODA*, 2002.

[18] P. B. Gibbons and Y. Matias. Synopsis Data Structures for Massive Data Sets. *Proc. 10th SODA* pp. 909–910, 1999.

[19] A. C. Gilbert, Y. Kotidis, S. Muthukrishnan, and M. J. Strauss. Surfing Wavelets on Streams: One-Pass Summaries for Approximate Aggregate Queries. *Proc. 27th VLDB* pp. 79-88, 2001.

[20] A. C. Gilbert, Y. Kotidis, S. Muthukrishnan, and M. J. Strauss. QuickSAND: Quick Summary and Analysis of Network Data. DIMACS Technical Report 2001-43, November 2001.

[21] A. C. Gilbert, S. Guha, P. Indyk, Y. Kotidis, S. Muthukrishnan, and M. J. Strauss. Fast, Small-Space Algorithms for Approximate Histogram Maintenance. *these proceedings.*

[22] A. Gionis, D. Gunopulos, and N. Koudas. Efficient

and Tunable Similar Set Retrieval. *Proc. SIGMOD Conference* 2001.

[23] A. Gionis, P. Indyk, and R. Motwani. Similarity Search in High Dimensions via Hashing. *Proc. 25th VLDB* pp. 518-529, 1999.

[24] M. X. Goemans and D. P. Williamson. Improved Approximation Algorithms for Maximum Cut and Satisfiability Problems Using Semidefinite Programming. *JACM* 42(6): 1115-1145, 1995.

[25] S. Guha, N. Mishra, R. Motwani, and L. O'Callaghan. Clustering data streams. *Proc. 41st FOCS*, pp. 359–366, 2000.

[26] A. Gupta and Éva Tardos. A constant factor approximation algorithm for a class of classification problems. *Proc. 32nd STOC*, pp. 652–658, 2000.

[27] T. H. Haveliwala, A. Gionis, and P. Indyk. Scalable Techniques for Clustering the Web. *Proc. 3rd WebDB*, pp. 129-134, 2000.

[28] P. Indyk. A small approximately min-wise independent family of hash functions. *Proc. 10th SODA*, pp. 454–456, 1999.

[29] P. Indyk. On approximate nearest neighbors in non-Euclidean spaces. *Proc. 40th FOCS*, pp. 148-155, 1999.

[30] P. Indyk. Stable Distributions, Pseudorandom Generators, Embeddings and Data Stream Computation. *Proc. 41st FOCS*, 189-197, 2000.

[31] Indyk, P., Motwani, R. Approximate nearest neighbors: towards removing the curse of dimensionality. *Proc. 30th STOC* pp. 604–613, 1998.

[32] P. Indyk, R. Motwani, P. Raghavan, and S. Vempala. Locality-Preserving Hashing in Multidimensional Spaces. *Proc. 29th STOC*, pp. 618–625, 1997.

[33] J. M. Kleinberg and Éva Tardos Approximation Algorithms for Classification Problems with Pairwise Relationships: Metric Labeling and Markov Random Fields. *Proc. 40th FOCS*, pp. 14–23, 1999.

[34] N. Linial and O. Sasson. Non-Expansive Hashing. *Combinatorica* 18(1): 121-132, 1998.

[35] N. Nisan. Pseudorandom sequences for space bounded computations. *Combinatorica*, 12:449–461, 1992.

[36] Y. Rubner. Perceptual Metrics for Image Database Navigation. Phd Thesis, Stanford University, May 1999

[37] Y. Rubner, L. J. Guibas, and C. Tomasi. The Earth Mover's Distance, Multi-Dimensional Scaling, and Color-Based Image Retrieval. *Proc. of the ARPA Image Understanding Workshop*, pp. 661-668, 1997.

[38] Y. Rubner, C. Tomasi. Texture Metrics. *Proc. IEEE International Conference on Systems, Man, and Cybernetics*, 1998, pp. 4601-4607.

[39] Y. Rubner, C. Tomasi, and L. J. Guibas. A Metric for Distributions with Applications to Image Databases. *Proc. IEEE Int. Conf. on Computer Vision*, pp. 59-66, 1998.

[40] Y. Rubner, C. Tomasi, and L. J. Guibas. The Earth Mover's Distance as a Metric for Image Retrieval. Tech. Report STAN-CS-TN-98-86, Dept. of Computer Science, Stanford University, 1998.

[41] M. Ruzon and C. Tomasi. Color edge detection with the compass operator. *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 2:160-166, 1999.

[42] M. Ruzon and C. Tomasi. Corner detection in textured color images. *Proc. IEEE Int. Conf. Computer Vision*, 2:1039-1045, 1999.