

Sign Stable Projections, Sign Cauchy Projections and Chi-Square Kernels

Ping Li

Department of Statistics & Biostatistics
Department of Computer Science
Rutgers University
New Brunswick, NJ 08901
pl314@rci.rutgers.edu

Gennady Samorodnitsky

School of ORIE and Department of Statistical Science
Cornell University
Ithaca, NY 14853
gs18@cornell.edu

John Hopcroft

Department of Computer Science
Cornell University
Ithaca, NY 14853
jeh@cs.cornell.edu

Abstract

The method of *stable random projections* is popular for efficiently computing the l_α distances in high dimension (where $0 < \alpha \leq 2$), using small space. Because it adopts nonadaptive linear projections, this method is naturally suitable when the data are collected in a dynamic streaming fashion (i.e., *turnstile* data streams). In this paper, we propose to use only the signs of the projected data and analyze the probability of collision (i.e., when the two signs differ). We derive a bound of the collision probability which is exact when $\alpha = 2$ and becomes less sharp when α moves away from 2. Interestingly, when $\alpha = 1$ (i.e., Cauchy random projections), we show that the probability of collision can be accurately approximated as functions of the chi-square (χ^2) similarity. For example, when the (un-normalized) data are binary, the maximum approximation error of the collision probability is smaller than 0.0192. In text and vision applications, the χ^2 similarity is a popular measure for nonnegative data when the features are generated from histograms. Our experiments confirm that the proposed method is promising for large-scale learning applications.

1 Introduction

High-dimensional representations have become very popular in modern applications of machine learning, computer vision, and information retrieval. Given two high-dimensional data vectors (e.g., two images or Web pages) $u, v \in \mathbb{R}^D$, a basic task is to compute their *distance* or *similarity*. For example, the correlation (ρ_2) and l_2 distance (d_2) are commonly used:

$$\rho_2(u, v) = \frac{\sum_{i=1}^D u_i v_i}{\sqrt{\sum_{i=1}^D u_i^2 \sum_{i=1}^D v_i^2}}, \quad d_2(u, v) = \sum_{i=1}^D |u_i - v_i|^2 \quad (1)$$

The definition of d_2 naturally extends to the l_α distance d_α :

$$d_\alpha(u, v) = \sum_{i=1}^D |u_i - v_i|^\alpha \quad (2)$$

Here, α can often be viewed as *tuning* parameter in practice. It is known that when $0 < \alpha \leq 2$, the commonly used (generalized) radial basis kernel defined as $K(u, v) = \exp(-\lambda d_\alpha(u, v))$, where $\lambda > 0$, is positive definite [26].

In this study, we are particularly interested in the χ^2 similarity, denoted by ρ_{χ^2} :

$$\rho_{\chi^2} = \sum_{i=1}^D \frac{2u_i v_i}{u_i + v_i}, \quad \text{where } u_i \geq 0, v_i \geq 0, \sum_{i=1}^D u_i = \sum_{i=1}^D v_i = 1 \quad (3)$$

The chi-square similarity is closely related to the chi-square distance d_{χ^2} :

$$d_{\chi^2} = \sum_{i=1}^D \frac{(u_i - v_i)^2}{u_i + v_i} = \sum_{i=1}^D (u_i + v_i) - \sum_{i=1}^D \frac{4u_i v_i}{u_i + v_i} = 2 - 2\rho_{\chi^2} \quad (4)$$

The chi-square similarity is an instance of the Hilbertian metrics, which are defined over probability space [11] and hence suitable for data generated from histograms. Histogram-based features (e.g., bag-of-word or bag-of-visual-word models) are extremely popular in computer vision, natural language processing (NLP), and information retrieval. Empirical studies have demonstrated the superiority of the χ^2 distance over l_2 or l_1 distances for image and text classification tasks [4, 11, 14, 2, 30, 29, 28].

The current trend in machine learning is to use ultra-high-dimensional representations. For example, Winner of 2009 PASCAL image classification challenge used 4 million features [31]. [1, 32] described applications using billion or trillion features. The use of extremely high-dimensional data often produces good accuracies but the cost is the significant increase in computations, storage, and energy consumptions.

The method of *normal random projections* (i.e., α -stable projections with $\alpha = 2$) has become popular in machine learning (e.g., [7, 19]) for reducing the data dimensions and data sizes, to facilitate efficient computations of the l_2 distances and correlations. More generally, the method of *stable random projections* [12, 18] provides an efficient algorithm to compute the l_α distances, for $0 < \alpha \leq 2$.

In this paper, we propose to use only the signs of the projected data after applying α -stable projections on the original data. In particular, we show that when $\alpha = 1$, the probability of collision (i.e., when the two signs differ) is closely related to the χ^2 similarity. Thus, our method provides an effective strategy for large-scale machine learning when the applications favor the use of χ^2 similarity.

1.1 Stable Random Projections and Sign (1-Bit) Stable Random Projections

Consider two high-dimensional vectors $u, v \in \mathbb{R}^D$. The basic idea of stable random projections is to multiply u and v by a random matrix $\mathbf{R} \in \mathbb{R}^{D \times k}$: $x = u\mathbf{R} \in \mathbb{R}^k$, $y = v\mathbf{R} \in \mathbb{R}^k$, where entries of \mathbf{R} are i.i.d. samples from a symmetric α -stable distribution with unit scale. By properties of stable distributions, $x_j - y_j$ follows a symmetric α -stable distribution with scale d_α . Hence, the task of computing d_α boils down to estimating the scale d_α from k i.i.d. samples.

In this paper, we propose to store only the signs of projected data and study the probability of collision:

$$P_\alpha = \Pr(\text{sign}(x_j) \neq \text{sign}(y_j)) \quad (5)$$

Using only the signs (i.e., 1 bit) has significant advantages for applications in search and learning. When $\alpha = 2$, this probability can be analytically evaluated [9] (or via a simple geometric argument):

$$P_2 = \Pr(\text{sign}(x_j) \neq \text{sign}(y_j)) = \frac{1}{\pi} \cos^{-1} \rho_2 \quad (6)$$

which is an important result known as *sim-hash* [5]. For $\alpha < 2$, the collision probability is an open problem. When the data are nonnegative, this paper will prove a bound of P_α for general $0 < \alpha \leq 2$:

$$P_\alpha = \Pr(\text{sign}(x) \neq \text{sign}(y)) \leq \frac{1}{\pi} \cos^{-1} \rho_\alpha, \quad \text{where } \rho_\alpha = \left(\frac{\sum_{i=1}^D u_i^{\alpha/2} v_i^{\alpha/2}}{\sqrt{\sum_{i=1}^D u_i^\alpha \sum_{i=1}^D v_i^\alpha}} \right)^{2/\alpha} \quad (7)$$

This bound is exact at $\alpha = 2$ and less sharp as α moves away from 2. Furthermore, for $\alpha = 1$ and nonnegative data, we have the interesting observation that the probability P_1 can be well approximated as functions of the χ^2 similarity ρ_{χ^2} .

1.2 The Advantages of Sign Stable Random Projections

The advantage of *sign stable random projections* can be summarized as follows:

1. There is a significant saving in storage space by using only 1 bit instead of (e.g.,) 64 bits.
2. This scheme leads to an efficient linear algorithm (e.g., linear SVM). For example, a negative sign can be coded as “01” and a positive sign as “10” (i.e., a vector of length 2). With k projections, we concatenate k short vectors to form a vector of length $2k$. This idea is inspired by *b-bit minwise hashing* [20, 21], which is designed for binary data and the “resemblance” similarity.
3. This scheme also leads to an efficient near neighbor search algorithm [8]. For this application, we can code a negative sign by “0” and positive sign by “1” and concatenate k such bits to form a hash table of 2^k buckets. In the query phase, one only searches for similar vectors in one bucket. Usually L such tables are independently constructed to increase the accuracy. This provides a simple and efficient implementation of the general framework of locality sensitive hashing [13]. For this paper, we will not focus on near neighbor search.

1.3 Data Stream Computations

Stable random projections are naturally suitable for *data stream* computations. In modern applications, massive datasets are often generated in a streaming fashion, which are difficult to transmit and store [23], as the processing is done **on the fly** in **one-pass** of the data. The problem of “scaling up for high dimensional

data and high speed data streams” is among the “ten challenging problems in data mining research” [33]. Network traffic is a typical example of data streams and network data can easily reach petabyte scale [33].

In the standard *turnstile* model [23], a data stream can be viewed as high-dimensional vector with the entry values changing over time. Here, we denote a stream at time t by $u_i^{(t)}$, $i = 1$ to D . At time t , a stream element (i_t, I_t) arrives and updates the i_t -th coordinate as

$$u_{i_t}^{(t)} = u_{i_t}^{(t-1)} + I_t \quad (8)$$

Clearly, the turnstile data stream model is particularly suitable for describing histograms and it is also a standard model for network traffic summarization and monitoring [34]. Because this stream model is linear, methods based on linear projections (i.e., matrix-vector multiplications) can naturally handle streaming data of this sort. Basically, entries of the projection matrix $\mathbf{R} \in \mathbb{R}^{D \times k}$ are (re)generated as needed using pseudo-random number techniques [24]. At the element (i_t, I_t) arrives, only the entries in the i_t -th row, i.e., $r_{i_t, j}$, $j = 1$ to k , are (re)generated and the projected data are updated as

$$x_j^{(t)} = x_j^{(t-1)} + I_t \times r_{i_t, j}, \quad j = 1, 2, \dots, D \quad (9)$$

Thus, the original data, e.g., $u, v \in \mathbb{R}^D$, are not stored, as a significant advantage of using linear projections.

Recall that, in the definition of χ^2 similarity, the data are assumed to be normalized (summing to 1). For nonnegative data streams, the sum can be computed error-free by using merely one counter: $\sum_{i=1}^D u_i^{(t)} = \sum_{s=1}^t I_s$. This means that we can still use, without loss of generality, the sum-to-one assumption, even in the streaming environment. This fact was recently exploited by another data stream algorithm named *Compressed Counting (CC)* [22] for efficiently estimating the *Shannon entropy* of data streams.

Because the χ^2 similarity is very popular in computer vision, recently there are proposals for estimating the χ^2 similarity. For example, [16] proposed a nice technique to approximate ρ_{χ^2} by first expanding the data from D dimensions to (e.g.,) $5 \sim 10 \times D$ dimensions through a *nonlinear* transformation and then applying normal random projections on the expanded data. Because the nonlinear operations must be applied to the original data, the method is not applicable to data stream computations.

For the sake of notational simplicity, we will drop the superscript (t) in the rest of the paper.

1.4 Paper Organization

In Section 2, we provide a set of SVM experiments to illustrate the potential advantages of the use of χ^2 similarity ρ_{χ^2} compared to linear kernels. For example, on the *MNIST-small* dataset, using χ^2 kernels can achieve significantly more accurate classification results than using linear kernels, without using additional tuning parameter.

In Section 3, we derive a general bound of the collision probability for sign stable random projections, for all $0 < \alpha \leq 2$. As verified by simulations, this bound is exact when $\alpha = 2$ and becomes less sharp as α moves away from 2. In Section 4, we focus on Cauchy random projections (i.e., $\alpha = 1$) and analyze several properties of the χ^2 similarity.

In Section 5, we propose two approximations of the collision probability for $\alpha = 1$, as functions of the χ^2 similarity. We prove that when the data are binary, the approximation error would be smaller than 0.0192. Furthermore, we validate the proposed approximations on a dataset of 3.6 million pairs of word occurrence vectors. Section 6 provides an experimental study for SVM classifications using our proposed approximations. Finally, Section 7 concludes the paper and also points out that the processing cost of stable random projections can be substantially reduced by using a *very sparse* projection matrix.

2 An Experimental Study of Chi-Square Kernels

We provide an experimental study to validate the use of χ^2 similarity. Here, the “ χ^2 -kernel” is defined as $K(u, v) = \rho_{\chi^2}$ and the “acos- χ^2 -kernel” as $K(u, v) = 1 - \frac{1}{\pi} \cos^{-1} \rho_{\chi^2}$. With a slight abuse of terminology, we call both “ χ^2 kernel” when it is clear in the context.

We use the “precomputed kernel” functionality in LIBSVM on two datasets: (i) *UCI-PEMS*, with 267 training examples and 173 testing examples in 138,672 dimensions; (ii) *MNIST-small*, a subset of the popular *MNIST* dataset, with 10,000 training examples and 10,000 testing examples.¹

The results are shown in Figure 1. To compare these two types of χ^2 kernels with “linear” kernel, we also test the same data using LIBLINEAR [6] after normalizing the data to have unit Euclidian norm, i.e., we basically use ρ_2 . For both LIBSVM and LIBLINEAR, we use l_2 -regularization with a parameter C and we report the test errors for a wide range of C values.

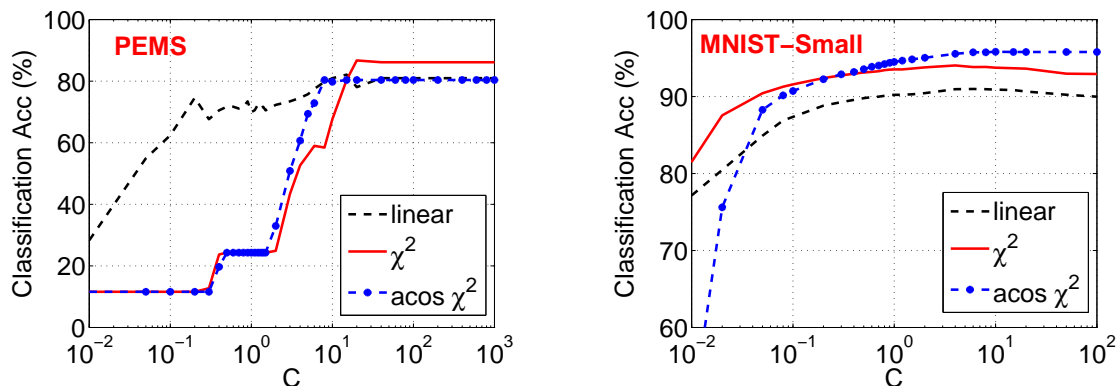


Figure 1: Classification accuracies. C is the l_2 -regularization parameter. We use LIBLINEAR for “linear” (i.e., ρ_2) kernel and LIBSVM “precomputed kernel” functionality for two types of χ^2 kernels (“ χ^2 -kernel” and “acos- χ^2 -kernel”). For *UCI-PEMS*, the χ^2 -kernel has better performance than the linear kernel and acos- χ^2 -kernel. For *MNIST-Small*, both χ^2 kernels noticeably outperform linear kernel. Note that, although we place three curves together, we do not mean one should compare them at a fixed C . We assume the best C could be found by (e.g.,) cross-validations for each method. The *MNIST-small* dataset used the original testing examples and only 1/6 of the original training examples.

Here, we should state that, while the use of χ^2 similarity has become popular in computer vision, it is not the intention of this paper to use these two small examples to conclude the advantage of χ^2 kernels over linear kernel. We simply use them to validate our method for approximating the χ^2 similarity, as our proposed method is general-purpose, not limited to data generated from histograms.

The *MNIST-small* dataset is more close to computer vision applications. Using linear kernels, the classification accuracy is about 90% at its best C value, while using the χ^2 kernels can achieve about 95%. Note that our proposed method will be able to approximate the acos- χ^2 -kernel by linear kernels. In other words, our method may be able to provide both the computational efficiency of linear kernels and good accuracy of χ^2 kernels (in certain datasets). Unlike the RBF nonlinear kernel, the χ^2 kernels do not use additional tuning parameters (other than C). Since *MNIST-small* used only 1/6 of the original training examples and the same testing examples, the performance of χ^2 kernels on this dataset is impressive.

¹When we used the original MNIST data (60,000 training examples) using LIBSVM “precomputed kernel” functionality (which is memory intensive), we observed seg fault in matlab on a workstation with 96GB RAM.

3 Sign Stable Random Projections and the Collision Probability Bound

Consider two high-dimensional vectors $u, v \in \mathbb{R}^D$. We apply stable random projections:

$$x = \sum_{i=1}^D u_i r_i, \quad y = \sum_{i=1}^D v_i r_i, \quad r_i \sim S(\alpha, 0, 1), \text{ i.i.d.} \quad (10)$$

Here $Z \sim S(\alpha, \beta, \gamma)$ denotes an α -stable β -skewed distribution with scale $\gamma > 0$. When $\beta = 0$ (i.e., symmetric), its characteristic function [25] is $E\left(e^{\sqrt{-1}Zt}\right) = e^{-\gamma|t|^\alpha}$. By properties of stable distributions, we know $x - y \sim S\left(\alpha, 0, \sum_{i=1}^D |u_i - v_i|^\alpha\right)$. This method transforms the task of computing the l_α distance $\sum_{i=1}^D |u_i - v_i|^\alpha$ into a problem of estimating the scale parameter from stable samples. Note that in this section, we only use a projection vector $\mathbf{R} \in \mathbb{R}^{D \times k}$ with $k = 1$, because the samples are i.i.d. anyway. Consequently, we simply use x and y to denote the projected instead of x_j and y_j .

Applications including linear learning and near neighbor search will benefit from *sign α -stable random projections*. When $\alpha = 2$, the collision probability $\Pr(\text{sign}(x) \neq \text{sign}(y))$ is known [5, 9]. For $\alpha < 2$, it is a difficult probability problem. In this section, we will study a bound of $\Pr(\text{sign}(x) \neq \text{sign}(y))$ for general $0 < \alpha \leq 2$, which is fairly accurate for α close to 2.

3.1 Collision Probability Bound

In this paper, we focus on nonnegative data (as common in practice). We present our first theorem.

Theorem 1 *When the data are nonnegative, $u_i \geq 0, v_i \geq 0$, we have*

$$\Pr(\text{sign}(x) \neq \text{sign}(y)) = E\left(\frac{1}{\pi} \cos^{-1}\left(\frac{\sum_{i=1}^D u_i v_i T_i}{\sqrt{\sum_{i=1}^D u_i^2 T_i \sum_{i=1}^D v_i^2 T_i}}\right)\right) \quad (11)$$

$$\leq \frac{1}{\pi} \cos^{-1}\left\{\left(\frac{\sum_{i=1}^D u_i^{\alpha/2} v_i^{\alpha/2}}{\sqrt{\sum_{i=1}^D u_i^\alpha \sum_{i=1}^D v_i^\alpha}}\right)^{2/\alpha}\right\} = \frac{1}{\pi} \cos^{-1} \rho_\alpha \quad (12)$$

where $T_i \sim S\left(\alpha/2, 1, \cos\left(\frac{\pi\alpha}{4}\right)\right)$, i.i.d.

Proof: See Appendix A. Recall ρ_α is also defined in (7). □

The proof of Theorem 1 utilizes an interesting property: a symmetric stable random variable can be factorized into a product of a normal and a maximally-skewed stable random variable [25].

Lemma 1 *Let $X \sim S(\alpha', 0, \gamma)$ with $0 < \alpha' \leq 2$, $0 < \alpha < \alpha'$, and $T \sim S(\alpha/\alpha', 1, \cos(\frac{\pi\alpha}{2\alpha'}))$. T and X are independent. Then*

$$Z = T^{1/\alpha'} X \sim S(\alpha, 0, \gamma) \quad (13)$$

In particular, when $\alpha' = 2$, we have

$$Z = T^{1/2} X \sim S(\alpha, 0, \gamma), \quad \text{where } T \sim S\left(\alpha/2, 1, \cos\left(\frac{\pi\alpha}{4}\right)\right), \quad X \sim N(0, 2\gamma) \quad (14)$$

□

How tight is this bound: $\Pr(\text{sign}(x) \neq \text{sign}(y)) \leq \frac{1}{\pi} \cos^{-1} \rho_\alpha$? The answer depends on α . For $\alpha = 2$, this bound is exact [5, 9]. In fact the result for $\alpha = 2$ immediately leads to the following Lemma:

Lemma 2 *The kernel defined as $K(u, v) = 1 - \frac{1}{\pi} \cos^{-1} \rho_2$ is positive definite (PD).*

Proof: *The indicator function $1\{\text{sign}(x) = \text{sign}(y)\}$ can be written as an inner product (hence PD) and $\Pr(\text{sign}(x) = \text{sign}(y)) = E(1\{\text{sign}(x) = \text{sign}(y)\}) = 1 - \frac{1}{\pi} \cos^{-1} \rho_2$. \square*

3.2 A Simulation Study to Verify the Bound of the Collision Probability

We generate the original data u and v by sampling from a bivariate t-distribution, which has two parameters: the correlation and the number of degrees of freedom. We use a full range of the correlation parameter from 0 to 1 (spaced at 0.01). To generate positive data, we simply take the absolute values of the generated data. Then we fix the data as our original data (like u and v), apply sign stable random projections, and report the empirical collision probabilities (after 10^5 repetitions). An important parameter for t-distribution is the number of degrees of freedom (df), which determines how heavy the tail is. For example, a t-distribution has finite expectation when $df > 1$ and has finite third moment when $df > 3$. We only report simulations with $df = 1$ (i.e., heavy-tailed data). The results for larger df values are similar in our experiments.

Figure 2 presents the simulated collision probability $\Pr(\text{sign}(x) \neq \text{sign}(y))$ for $D = 100$ and $\alpha \in \{1.8, 1.5, 1.2, 1.0, 0.5, 0.2\}$. In each panel, the dashed curve is the theoretical upper bound $\frac{1}{\pi} \cos^{-1} \rho_\alpha$, and the solid curve is the simulated collision probability. Note that it is expected that the simulated data can not cover the entire range of ρ_α values. For example, if the data are dense (no zero entries), then $\rho_\alpha \rightarrow 1$ as $\alpha \rightarrow 0$. This explains why the last panel ($\alpha = 0.2$) only contains data with $\rho_\alpha > 0.85$.

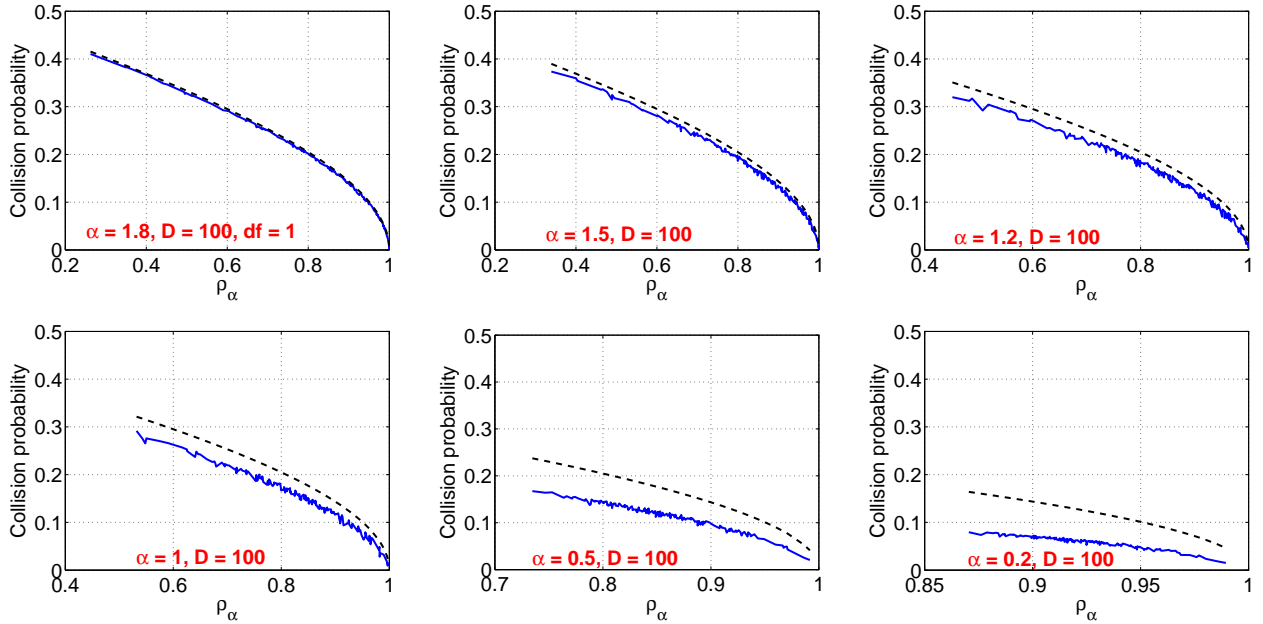


Figure 2: **Dense Data** and $D = 100$. Simulated collision probability $\Pr(\text{sign}(x) \neq \text{sign}(y))$ for sign stable random projections. In each panel, the dashed curve is the upper bound $\frac{1}{\pi} \cos^{-1} \rho_\alpha$.

Figure 2 verifies the theoretical upper bound $\frac{1}{\pi} \cos^{-1} \rho_\alpha$. When $\alpha \geq 1.5$, this upper bound is fairly sharp. However, when $\alpha \leq 1$, the bound is not tight, especially for small α . Also, the curves of the empirical collision probabilities are not smooth (in terms of ρ_α).

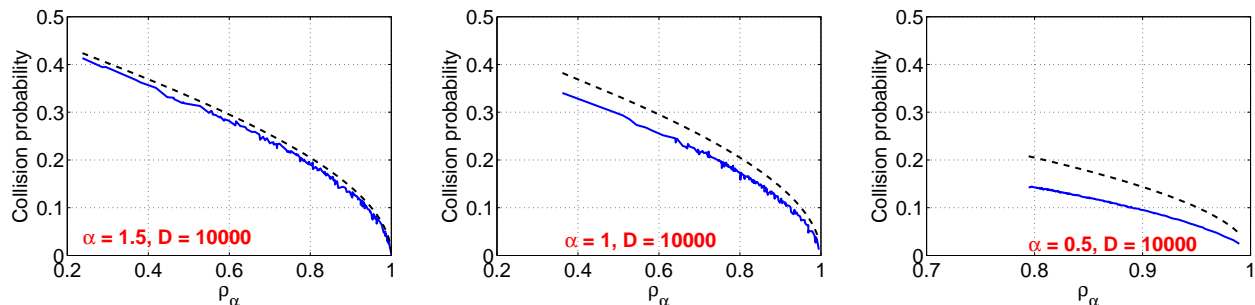


Figure 3: **Dense Data** and $D = 10,000$. Simulated collision probability $\Pr(\text{sign}(x) \neq \text{sign}(y))$ for sign stable random projections.

Real-world high-dimensional datasets are often **sparse**. To verify the theoretical upper bound of the collision probability on sparse data, we also simulate sparse data by randomly making 50% of the generated data used in Figure 2 be zero. With sparse data, it is even more obvious that the theoretical upper bound $\frac{1}{\pi} \cos^{-1} \rho_\alpha$ is not sharp when $\alpha \leq 1$, as shown in Figure 4.

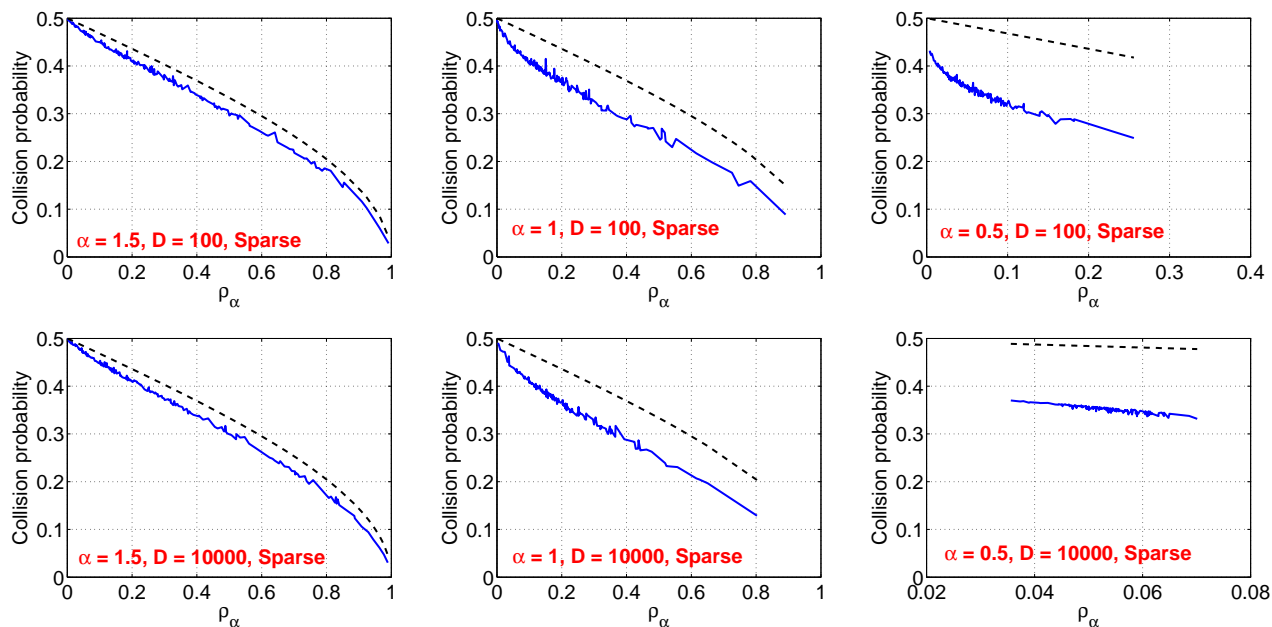


Figure 4: **Sparse Data**. Simulated collision probability $\Pr(\text{sign}(x) \neq \text{sign}(y))$ for sign stable random projection, for $D = 100$ (upper panels) and $D = 10,000$ (bottom panels). The upper bound is not tight for $\alpha \leq 1$.

In summary, the collision probability bound: $\Pr(\text{sign}(x) \neq \text{sign}(y)) \leq \frac{1}{\pi} \cos^{-1} \rho_\alpha$ is fairly sharp when α is close to 2 (e.g., $\alpha \geq 1.5$). However, for $\alpha \leq 1$, a better approximation is needed.

4 $\alpha = 1$ and Chi-Square (χ^2) Similarity

In this section, we focus on nonnegative data ($u_i \geq 0, v_i \geq 0$) and $\alpha = 1$. This case is important in practice. For example, we can view the data (u_i, v_i) as empirical probabilities, which are common when the data are generated from histograms (a popular technique in NLP and computer vision) [4, 11, 14, 2, 30, 29, 28].

In this context, we always normalize the data, i.e., $\sum_{i=1}^D u_i = \sum_{i=1}^D v_i = 1$. From the collision probability bound in Theorem 1, we know that

$$\Pr(\text{sign}(x) \neq \text{sign}(y)) \leq \frac{1}{\pi} \cos^{-1} \rho_1, \quad \text{where} \quad (15)$$

$$\rho_1 = \left(\frac{\sum_{i=1}^D u_i^{1/2} v_i^{1/2}}{\sqrt{\sum_{i=1}^D u_i \sum_{i=1}^D v_i}} \right)^2 = \left(\sum_{i=1}^D u_i^{1/2} v_i^{1/2} \right)^2 \quad (16)$$

This bound is not tight, as illustrated by our simulation study. Interestingly, this collision probability can be related to the χ^2 similarity.

Recall the definitions of the chi-square distance $d_{\chi^2} = \sum_{i=1}^D \frac{(u_i - v_i)^2}{u_i + v_i}$ and the chi-square similarity $\rho_{\chi^2} = 1 - \frac{1}{2} d_{\chi^2} = \sum_{i=1}^D \frac{2u_i v_i}{u_i + v_i}$. In this context, we should view $\frac{0}{0} = 0$.

Lemma 3 Assume $u_i \geq 0, v_i \geq 0, \sum_{i=1}^D u_i = 1, \sum_{i=1}^D v_i = 1$. Then

$$\rho_{\chi^2} = \sum_{i=1}^D \frac{2u_i v_i}{u_i + v_i} \geq \rho_1 = \left(\sum_{i=1}^D u_i^{1/2} v_i^{1/2} \right)^2 \quad (17)$$

Proof: By the Cauchy-Schwarz inequality

$$\begin{aligned} \sum_{i=1}^D u_i^{1/2} v_i^{1/2} &= \sum_{i=1}^D \frac{u_i^{1/2} v_i^{1/2}}{[(u_i + v_i)/2]^{1/2}} \left(\frac{u_i + v_i}{2} \right)^{1/2} \\ &\leq \left(\sum_{i=1}^D \frac{2u_i v_i}{u_i + v_i} \right)^{1/2} \left(\sum_{i=1}^D \frac{u_i + v_i}{2} \right)^{1/2} = \left(\sum_{i=1}^D \frac{2u_i v_i}{u_i + v_i} \right)^{1/2} \cdot 1 = \left(\sum_{i=1}^D \frac{2u_i v_i}{u_i + v_i} \right)^{1/2} \end{aligned}$$

□

It is known that the χ^2 -kernel ρ_{χ^2} is PD [11]. As a result, we know that the $\text{acos-}\chi^2$ -kernel is PD. The proof is analogous to the proof of Lemma 2.

Lemma 4 The kernel defined as $K(u, v) = 1 - \frac{1}{\pi} \cos^{-1} \rho_{\chi^2}$ is positive definite (PD).

Proof: Because ρ_{χ^2} is PD, we can write it as an inner product $\rho_{\chi^2} = \sum_i \tilde{u}_i \tilde{v}_i$, even though we do not know \tilde{u}_i and \tilde{v}_i explicitly. When the two original vectors u and v are identical, we have $\rho_{\chi^2} = 1$, which means we must have $\sum_i \tilde{u}_i^2 = \sum_i \tilde{v}_i^2 = 1$. At this point, we can apply normal random projections $x = \sum_i \tilde{u}_i r_i, y = \sum_i \tilde{v}_i r_i$, where $r_i \sim N(0, 1)$, i.i.d. Then, just like Lemma 2, the desired result follows from the properties of indicator function and expectation. Note that there is also an equivalent argument in terms of a Gaussian process. Since ρ_{χ^2} is PD, there exists a Gaussian process with covariance ρ_{χ^2} , whose variance (i.e., when $u_i = v_i$) is exactly $1 = \sum_{i=1}^D u_i = \sum_{i=1}^D \frac{2u_i u_i}{u_i + u_i}$. □

The remaining question is how to connect Cauchy random projections with the χ^2 similarity.

5 Two Approximations of Collision Probability for Sign Cauchy Projections

It is a difficult problem to derive the collision probability of sign Cauchy projections, especially if we would like to express the probability only in terms of certain summary statistics. Our **first** observation is that the collision probability can be well approximated using the χ^2 similarity:

$$\Pr(\text{sign}(x) \neq \text{sign}(y)) \approx P_{\chi^2(1)} = \frac{1}{\pi} \cos^{-1}(\rho_{\chi^2}) \quad (18)$$

Figure 5 shows this approximation is better than $\frac{1}{\pi} \cos^{-1}(\rho_1)$. In sparse data, the approximation is very accurate while the bound $\frac{1}{\pi} \cos^{-1}(\rho_1)$ is not sharp (and the curve is not smooth in ρ_1).

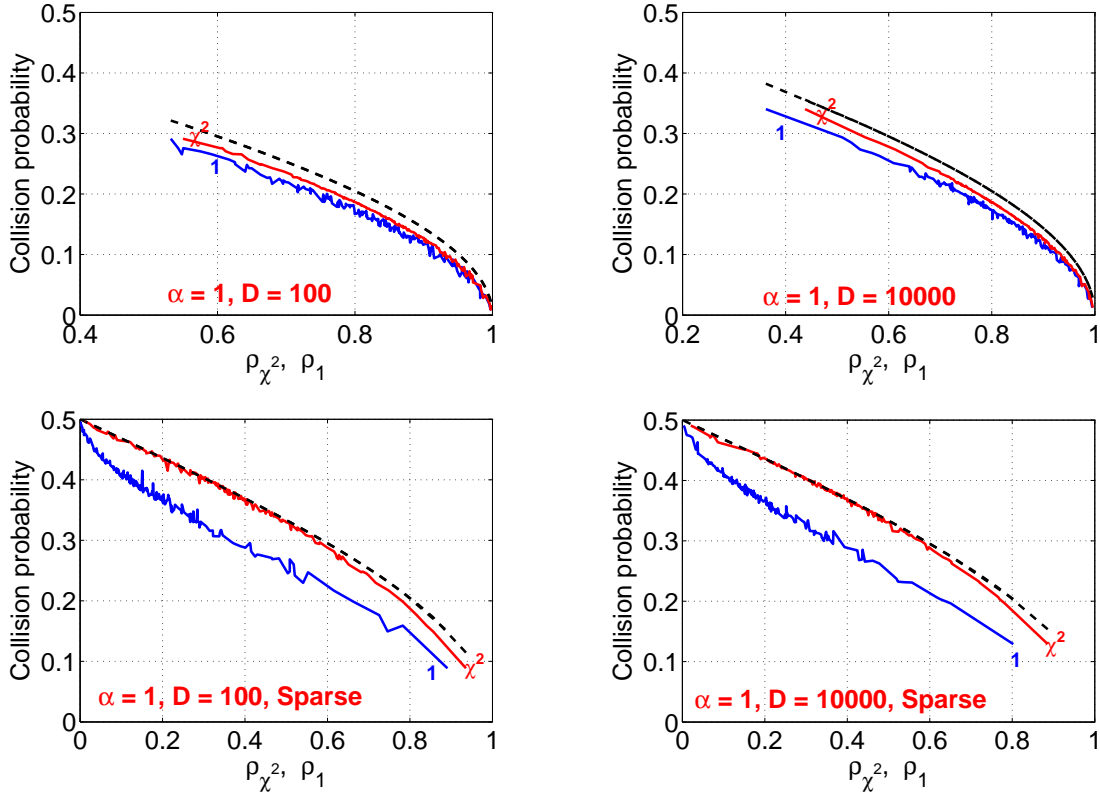


Figure 5: The dashed curve is $\frac{1}{\pi} \cos^{-1}(\rho)$, where ρ can be ρ_1 or ρ_{χ^2} depending on the context. In each panel, the two solid curves are the empirical collision probabilities in terms of ρ_1 (labeled by “1”) or ρ_{χ^2} (labeled by “ χ^2 ”). It is clear that the proposed approximation $\frac{1}{\pi} \cos^{-1} \rho_{\chi^2}$ in (18) is more tight than the upper bound $\frac{1}{\pi} \cos^{-1} \rho_1$, especially so in sparse data.

Our **second** (and less obvious) approximation is the following integral:

$$\Pr(\text{sign}(x) \neq \text{sign}(y)) \approx P_{\chi^2(2)} = \frac{1}{2} - \frac{2}{\pi^2} \int_0^{\pi/2} \tan^{-1} \left(\frac{\rho_{\chi^2}}{2 - 2\rho_{\chi^2}} \tan t \right) dt \quad (19)$$

Both approximations, $P_{\chi^2(1)}$ and $P_{\chi^2(2)}$, are monotone functions of ρ_{χ^2} . We can check the two boundary cases. When $\rho_{\chi^2} = 0$ (i.e., orthogonal data), we have $P_{\chi^2(1)} = P_{\chi^2(2)} = 1/2$ (i.e., the chance of collision is random). When $\rho_{\chi^2} = 1$ (i.e., identical data), we have $P_{\chi^2(1)} = P_{\chi^2(2)} = 0$ (i.e., two signs always equal).

In practice (e.g., near neighbor search or classification), we normally do not need the ρ_{χ^2} values explicitly. It usually suffices if the collision probability is a monotone function of the similarity. For theoretical analysis, of course, it is desirable to know the explicit expressions.

Figure 6 illustrates that, for dense data, second approximation (19) is more accurate than the first approximation (18). The second approximation (19) is also accurate for sparse data in this set of simulations.

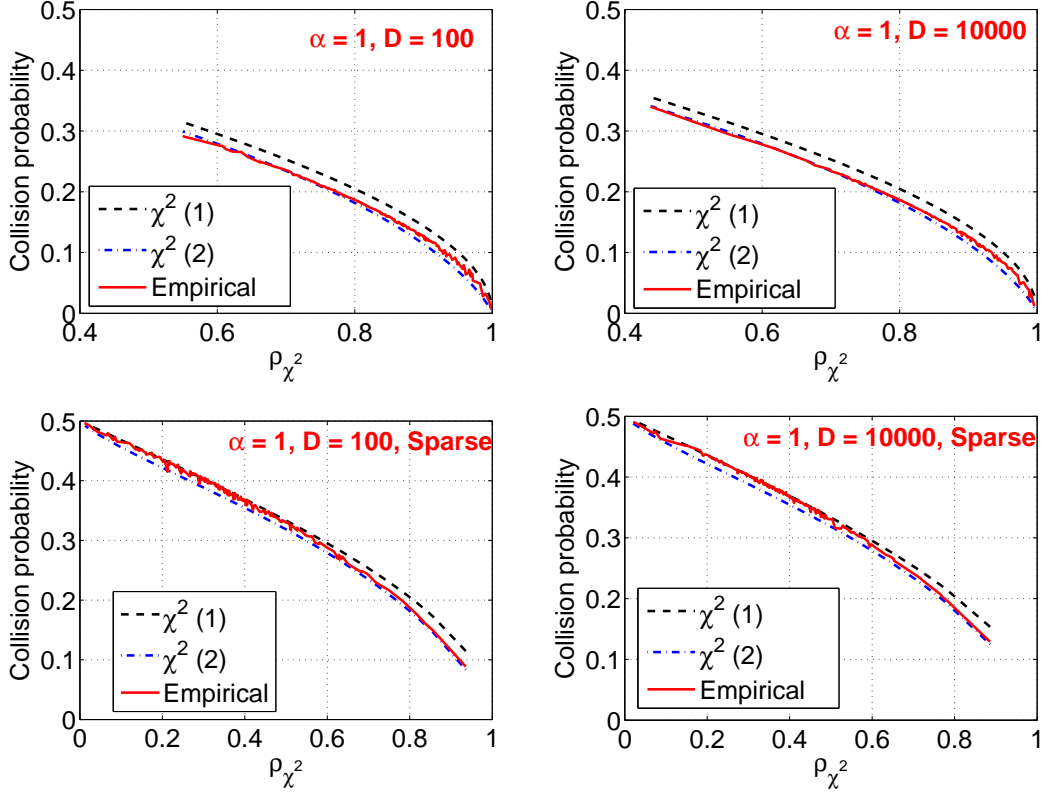


Figure 6: Comparison of two approximations: $\chi^2(1)$ based on (18) and $\chi^2(2)$ based on (19). The solid curves (empirical probabilities expressed in terms of ρ_{χ^2}) are the same solid curves labeled “ χ^2 ” in Figure 5. The top panels show that the second approximation (19) is more accurate in dense data. The bottom panels illustrate that both approximations are accurate in sparse data. (18) is slightly more accurate at small ρ_{χ^2} and (19) is more accurate at ρ_{χ^2} close to 1.

Moreover, Figure 7 shows that $P_{\chi^2(2)} \leq P_{\chi^2(1)}$ (as proved in Lemma 5)

Lemma 5

$$P_{\chi^2(2)} \leq P_{\chi^2(1)} \tag{20}$$

Proof: For $0 \leq z \leq 1$, to show

$$g(z) = \frac{1}{\pi} \cos^{-1}(z) + \frac{2}{\pi^2} \int_0^{\pi/2} \tan^{-1}\left(\frac{z}{2-2z} \tan t\right) dt - \frac{1}{2} \geq 0$$

Note that $g(0) = 0$, $g(1) = 0$, and

$$\begin{aligned} g'(z) &= -\frac{1}{\pi} \frac{1}{\sqrt{1-z^2}} + \frac{1}{\pi^2} \int_0^{\pi/2} \frac{\frac{1}{(1-z)^2} \tan t}{1 + \left(\frac{z}{2-2z} \tan t\right)^2} dt \\ &= -\frac{1}{\pi} \frac{1}{\sqrt{1-z^2}} + \frac{2}{\pi^2} \frac{1}{(z^2 - 4(1-z)^2)} \log \frac{z^2}{4(1-z)^2} \end{aligned}$$

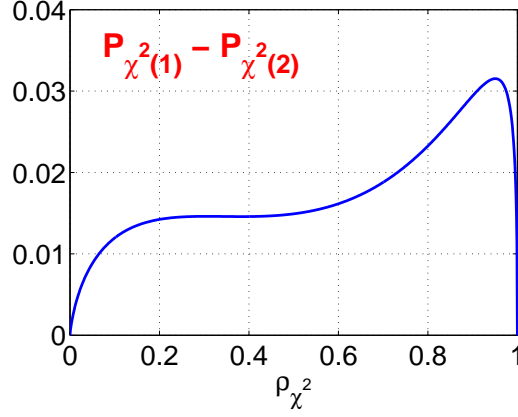


Figure 7: The difference of the two χ^2 approximations: $P_{\chi^2(1)} - P_{\chi^2(2)}$.

There are three extremum points (where $g'(z) = 0$), at $z_1 = 0.30555$, $z_2 = 0.38291$, and $z_3 = 0.95094$ (keeping five digits). We can check $g(z) > 0$ at all three extremum points. This completes the proof. \square

5.1 Binary Data

Interestingly, when the data are binary (before normalization), we can compute the collision probability exactly, which allows us to analytically assess the accuracy of the approximations. In fact, this case inspired us to propose the second approximation (19), which is otherwise not intuitive.

For convenience, we define $a = |I_a|$, $b = |I_b|$, $c = |I_c|$, where

$$I_a = \{i | u_i > 0, v_i = 0\}, \quad I_b = \{i | v_i > 0, u_i = 0\}, \quad I_c = \{i | u_i > 0, v_i > 0\}, \quad (21)$$

Assume binary data (before normalization, i.e., sum to one). That is,

$$u_i = \frac{1}{|I_a| + |I_c|} = \frac{1}{a + c}, \quad \forall i \in I_a \cup I_c, \quad v_i = \frac{1}{|I_b| + |I_c|} = \frac{1}{b + c}, \quad \forall i \in I_b \cup I_c \quad (22)$$

The chi-square similarity ρ_{χ^2} becomes $\rho_{\chi^2} = \sum_{i=1}^D \frac{2u_i v_i}{u_i + v_i} = \frac{2c}{a+b+2c}$ and hence $\frac{\rho_{\chi^2}}{2-2\rho_{\chi^2}} = \frac{c}{a+b}$.

Theorem 2 Assume binary data. When $\alpha = 1$, the exact collision probability is

$$\Pr(\text{sign}(x) \neq \text{sign}(y)) = \frac{1}{2} - \frac{2}{\pi^2} E \left\{ \tan^{-1} \left(\frac{c}{a} |R| \right) \tan^{-1} \left(\frac{c}{b} |R| \right) \right\} \quad (23)$$

where R is a standard Cauchy random variable.

Proof: See Appendix B. \square

When $a = 0$ or $b = 0$, we have $E \left\{ \tan^{-1} \left(\frac{c}{a} |R| \right) \tan^{-1} \left(\frac{c}{b} |R| \right) \right\} = \frac{\pi}{2} E \left\{ \tan^{-1} \left(\frac{c}{a+b} |R| \right) \right\}$. This observation inspires us to propose the approximation (19):

$$P_{\chi^2(2)} = \frac{1}{2} - \frac{1}{\pi} E \left\{ \tan^{-1} \left(\frac{c}{a+b} |R| \right) \right\} = \frac{1}{2} - \frac{2}{\pi^2} \int_0^{\pi/2} \tan^{-1} \left(\frac{c}{a+b} \tan t \right) dt$$

To validate this approximation for binary data, we study the difference between (23) and (19), i.e.,

$$\begin{aligned} Err &= \Pr(\text{sign}(x) \neq \text{sign}(y)) - P_{\chi^2(2)} = Z(a/c, b/c) \\ &= -\frac{2}{\pi^2} E \left\{ \tan^{-1} \left(\frac{1}{a/c} |R| \right) \tan^{-1} \left(\frac{1}{b/c} |R| \right) \right\} + \frac{1}{\pi} E \left\{ \tan^{-1} \left(\frac{1}{a/c + b/c} |R| \right) \right\} \end{aligned} \quad (24)$$

(24) can be easily computed by simulations. Figure 8 confirms that the errors are very small and larger than zero. The maximum error is smaller than 0.0192.

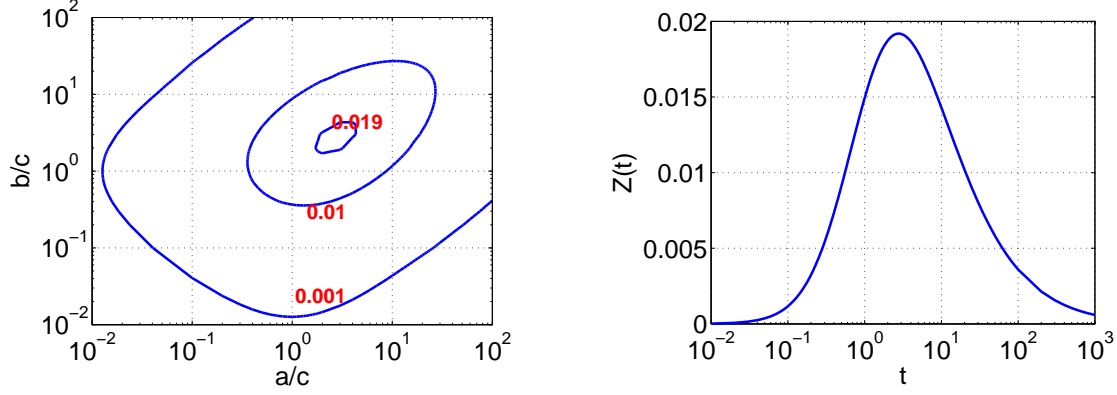


Figure 8: Left panel: contour plot for the error $Z(a/c, b/c)$ in (24). The maximum error (which is < 0.0192) occurs along the diagonal line. Right panel: the diagonal curve of $Z(a/c, b/c)$.

More precisely, we have the following Lemma.

Lemma 6 *The error defined in (24) ranges between 0 and $Z(t^*)$:*

$$0 \leq Z(a/c, b/c) \leq Z(t^*) = \int_0^\infty \left\{ -\frac{2}{\pi^2} \left(\tan^{-1} \left(\frac{r}{t^*} \right) \right)^2 + \frac{1}{\pi} \tan^{-1} \left(\frac{r}{2t^*} \right) \right\} \frac{2}{\pi} \frac{1}{1+r^2} dr \quad (25)$$

where $t^* = 2.77935$ is the solution to $\frac{1}{t^2-1} \log \frac{2t}{1+t} - \frac{\log(2t)}{(2t)^2-1} = 0$. Numerically, $Z(t^*) = 0.01919$.

Proof: See Appendix C. We keep five digits for the numerical values. □

5.2 An Experiment Based on 3.6 Million English Word Pairs

To further validate the two χ^2 approximations, we experiment with a word occurrences dataset (which is an example of histogram data) from a chunk of $D = 2^{16}$ web crawl documents. There are in total 2,702 words, i.e., 2,702 vectors and 3,649,051 word pairs. The entries of a vector are the occurrences of the word. This is a typical sparse, non-binary dataset. Interestingly, the errors of the collision probabilities based on two χ^2 approximations are still very small. To report the results, we apply sign Cauchy random projections 10^7 times to evaluate the approximation errors of (18) and (19).

The results, as presented in Figure 9, again confirm that the upper bound $\frac{1}{\pi} \cos^{-1} \rho_1$ is not tight and both χ^2 approximations, $P_{\chi^2(1)}$ and $P_{\chi^2(2)}$, are accurate.

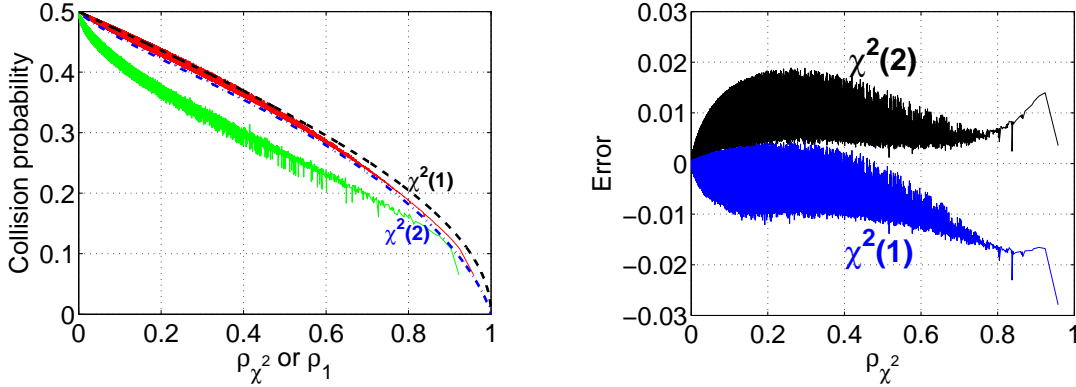


Figure 9: We compute the empirical collision probabilities for all 3.6 million English word pairs. In the **left panel**, we plot the empirical collision probabilities against ρ_1 (lower, green if color is available) and ρ_{χ^2} (higher, red). The curves confirm that the bound $\frac{1}{\pi} \cos^{-1} \rho_1$ is not tight (and the curve is not smooth). We plot the two χ^2 approximations as dashed curves which largely match the empirical collision probabilities plotted against ρ_{χ^2} . This confirms that the χ^2 approximations are good. For smaller ρ_{χ^2} values, the first approximation $P_{\chi^2(1)}$ is slightly more accurate. For larger ρ_{χ^2} values, the second approximation $P_{\chi^2(2)}$ is more accurate. In the **right panel**, we plot the errors for both $P_{\chi^2(1)}$ and $P_{\chi^2(2)}$.

6 Sign Cauchy Random Projections for Classification

Our method provides an effective strategy for classification. For each (high-dimensional) data vector, using k sign Cauchy projections, we encode a negative sign as “01” and a positive as “10” (i.e., a vector of length 2) and concatenate k short vectors to form a new feature vector of length $2k$ (with exactly k 1’s). We then feed the new dataset into a linear classifier (e.g., LIBLINEAR). Interestingly, this linear classifier approximates a nonlinear kernel classifier based on the $\text{acos-}\chi^2$ -kernel: $K(u, v) = 1 - \frac{1}{\pi} \cos^{-1} \rho_{\chi^2}$. See Figure 10 for the experiments on the same two datasets in Figure 1: *UCI-PEMS* and *MNIST-Small*.

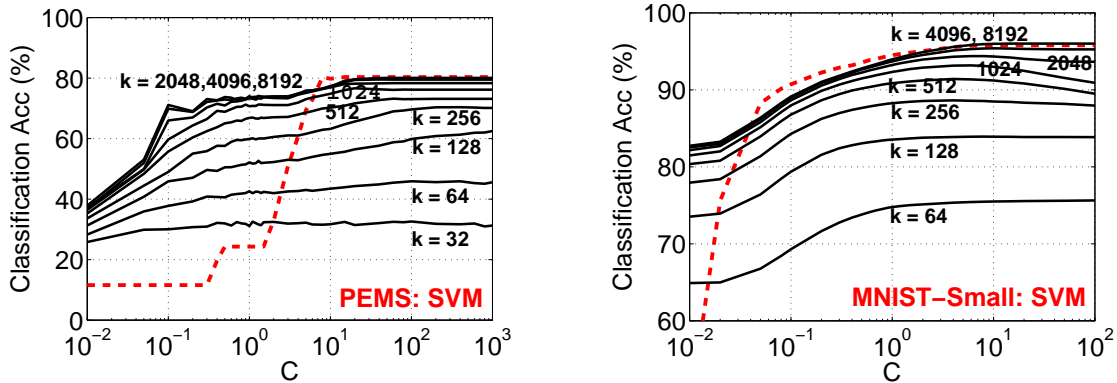


Figure 10: The two dashed (red if color is available) curves are the classification results obtained using “ $\text{acos-}\chi^2$ -kernel” via the “precomputed kernel” functionality in LIBSVM. The solid (black) curves are the accuracies using k sign Cauchy projections and LIBLINEAR. The results confirm that the linear kernel from sign Cauchy projections can approximate the nonlinear $\text{acos-}\chi^2$ -kernel.

Figure 1 has already shown that, for the *UCI-PEMS* dataset, the χ^2 -kernel (ρ_{χ^2}) can produce noticeably better classification results than the $\text{acos-}\chi^2$ -kernel ($1 - \frac{1}{\pi} \cos^{-1} \rho_{\chi^2}$). Although our method does

not directly approximate ρ_{χ^2} , we can still estimate ρ_{χ^2} by assuming the collision probability is exactly $\Pr(\text{sign}(x) \neq \text{sign}(y)) = \frac{1}{\pi} \cos^{-1} \rho_{\chi^2}$ and then we can feed the estimated ρ_{χ^2} values into LIBSVM “pre-computed kernel” for classification. Figure 11 verifies that this method can also approximate the χ^2 kernel with enough projections.

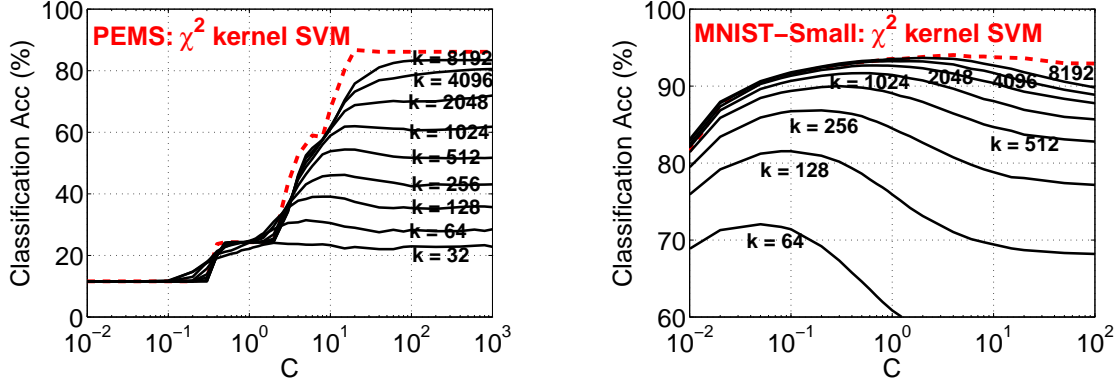


Figure 11: **Nonlinear kernels.** The dashed curves are the classification results obtained using χ^2 -kernel and LIBSVM “precomputed kernel” functionality. We apply k sign Cauchy projections and estimate ρ_{χ^2} assuming the collision probability is exactly $\frac{1}{\pi} \cos^{-1} \rho_{\chi^2}$ and then feed the estimated ρ_{χ^2} into LIBSVM again using the “precomputed kernel” functionality.

7 Conclusion

The use of χ^2 similarity is widespread in many machine learning applications, especially when features are generated from histograms, as common in natural language processing and computer vision. In fact, many prior studies [4, 11, 14, 2, 30, 29, 28] have shown the advantage of using χ^2 similarity compared to other measures such as l_2 or l_1 distances. Therefore, efficient algorithms for computing χ^2 similarities will be of great interest in machine learning, computer vision, and information retrieval.

For large-scale applications with ultra-high-dimensional datasets, using the χ^2 similarity becomes challenging for various practical reasons. Simply storing (and maneuvering) all the high-dimensional features would be difficult if there are a large number of observations (data points). Computing all pairwise χ^2 similarities can be time-consuming and in fact we usually can not materialize an all-pairwise similarity matrix if there are (e.g.,) merely 10^6 data points. Furthermore, the χ^2 similarity is nonlinear, making it difficult to take advantage of modern linear algorithms which are known to be very efficient, for example, [15, 27, 6, 3]. When data are generated in a streaming fashion, computing χ^2 similarities without storing the original data will be even more challenging.

The method of *stable random projections* [12, 18] is popular for efficiently computing the l_α distances in massive (and possibly streaming) data. We propose *sign stable random projections* by storing only the signs (i.e., 1-bit) of the projected data instead of real-valued numbers. The saving in storage would be a significant advantage. Also, the bits offer the indexing capability which allows efficient search. For example, we can build hash tables using the bits to achieve sublinear time near neighbor search, although this paper does not focus on near neighbor search. We can also build efficient linear classifiers using these bits, for large-scale high-dimensional applications with *Big Data*.

To analyze sign stable random projections, a crucial task is to study the probability of collision (i.e., when the two signs differ). For general $0 < \alpha \leq 2$, we derive a theoretical bound of the collision probability

which is exact when $\alpha = 2$. The bound is fairly sharp for α close to 2 (e.g., $\alpha \geq 1.5$). For $\alpha = 1$ (i.e., Cauchy random projections), we find the χ^2 approximation is significantly more accurate. In addition, for binary data, we can analytically show that the errors from using the χ^2 approximation are less than 0.0192. Experiments on real and simulated data confirm that our proposed χ^2 approximations are very accurate.

We are enthusiastic about the practicality of *sign stable projections* in large-scale learning and search applications. The previous idea of using the signs from *normal random projections* has been widely adopted in practice, for approximating correlations. Given the wide spread use of χ^2 similarity, the practical advantage of χ^2 similarity in text and vision applications, and the simplicity of our method, we expect the proposed method will be adopted by practitioners.

Finally, to conclude this paper, we should mention that the processing cost of conducting stable random projections can be dramatically reduced by *very sparse stable random projections* [17]. This will make our proposed method even more practical.

A Proof of Theorem 1

Apply stable random projections: $x = \sum_{i=1}^D u_i r_i$, $y = \sum_{i=1}^D v_i r_i$, $r_i \sim S(\alpha, 0, 1)$ i.i.d. Using the result in Lemma 1, we can factor r_i into a product of two independent random variables:

$$r_i = T_i^{1/2} \times N_i, \quad T_i \sim S\left(\alpha/2, 1, \cos\left(\frac{\pi\alpha}{4}\right)\right), \quad N_i \sim S(2, 0, 1)$$

Conditioning on T_1, T_2, \dots, T_D , the collision probability becomes

$$\Pr(\text{sign}(x) \neq \text{sign}(y) | T_i, i = 1, 2, \dots, D) = \frac{1}{\pi} \cos^{-1} \left(\frac{\sum_{i=1}^D u_i v_i T_i}{\sqrt{\sum_{i=1}^D u_i^2 T_i \sum_{i=1}^D v_i^2 T_i}} \right)$$

and hence the unconditional probability can be written as

$$\Pr(\text{sign}(x) \neq \text{sign}(y)) = E \left(\frac{1}{\pi} \cos^{-1} \left(\frac{\sum_{i=1}^D u_i v_i T_i}{\sqrt{\sum_{i=1}^D u_i^2 T_i \sum_{i=1}^D v_i^2 T_i}} \right) \right)$$

When the data are nonnegative, i.e., $u_i \geq 0, v_i \geq 0$, we have

$$\begin{aligned} \sum_{i=1}^D u_i v_i T_i &= \left(\sum_{i=1}^D u_i^{\alpha/2} v_i^{\alpha/2} \cos\left(\frac{\pi\alpha}{4}\right) \right)^{2/\alpha} \times S_1 \\ \sum_{i=1}^D u_i^2 T_i &= \left(\sum_{i=1}^D u_i^\alpha \cos\left(\frac{\pi\alpha}{4}\right) \right)^{2/\alpha} \times S_2, \\ \sum_{i=1}^D v_i^2 T_i &= \left(\sum_{i=1}^D v_i^\alpha \cos\left(\frac{\pi\alpha}{4}\right) \right)^{2/\alpha} \times S_3 \end{aligned}$$

where the random variables $S_1, S_2, S_3 \sim S(\alpha/2, 1, 1)$ are not independent.

As $\cos^{-1}(z)$ is a concave, monotonically decreasing function of $z \in [0, 1]$, we can bound the probability by Jensen's inequality as follows:

$$\begin{aligned}
& \Pr(\text{sign}(x) \neq \text{sign}(y)) \\
&= E \left(\frac{1}{\pi} \cos^{-1} \left(\frac{\sum_{i=1}^D u_i v_i T_i}{\sqrt{\sum_{i=1}^D u_i^2 T_i \sum_{i=1}^D v_i^2 T_i}} \right) \right) \\
&\leq \frac{1}{\pi} \cos^{-1} \left\{ E \left(\frac{\sum_{i=1}^D u_i v_i T_i}{\sqrt{\sum_{i=1}^D u_i^2 T_i \sum_{i=1}^D v_i^2 T_i}} \right) \right\} \\
&= \frac{1}{\pi} \cos^{-1} \left\{ E \exp \left(\log \left(\frac{\sum_{i=1}^D u_i v_i T_i}{\sqrt{\sum_{i=1}^D u_i^2 T_i \sum_{i=1}^D v_i^2 T_i}} \right) \right) \right\} \\
&\leq \frac{1}{\pi} \cos^{-1} \left\{ \exp \left(E \log \left(\frac{\sum_{i=1}^D u_i v_i T_i}{\sqrt{\sum_{i=1}^D u_i^2 T_i \sum_{i=1}^D v_i^2 T_i}} \right) \right) \right\} \\
&= \frac{1}{\pi} \cos^{-1} \left\{ \exp \left(\frac{2}{\alpha} \log \left(\sum_{i=1}^D u_i^{\alpha/2} v_i^{\alpha/2} \right) - \frac{1}{\alpha} \log \left(\sum_{i=1}^D u_i^\alpha \right) - \frac{1}{\alpha} \log \left(\sum_{i=1}^D v_i^\alpha \right) + E \log \frac{S_1}{\sqrt{S_2 S_3}} \right) \right\} \\
&= \frac{1}{\pi} \cos^{-1} \left\{ \left(\frac{\sum_{i=1}^D u_i^{\alpha/2} v_i^{\alpha/2}}{\sqrt{\sum_{i=1}^D u_i^\alpha \sum_{i=1}^D v_i^\alpha}} \right)^{2/\alpha} \right\} = \frac{1}{\pi} \cos^{-1} \rho_\alpha
\end{aligned}$$

Note that $E \log \frac{S_1}{\sqrt{S_2 S_3}} = E \log S_1 - 1/2 E \log S_2 - 1/2 E \log S_3 = 0$ because S_1, S_2, S_3 follow the same distribution (although they are not independent). This completes the proof.

B Proof of Theorem 2

For convenience, we define $a = |I_a|$, $b = |I_b|$, $c = |I_c|$, where

$$I_a = \{i | u_i > 0, v_i = 0\}, \quad I_b = \{i | v_i > 0, u_i = 0\}, \quad I_c = \{i | u_i > 0, v_i > 0\},$$

Assume binary data (before normalization), i.e.,

$$u_i = \frac{1}{|I_a| + |I_c|} = \frac{1}{a + c}, \quad \forall i \in I_a \cup I_c, \quad v_i = \frac{1}{|I_b| + |I_c|} = \frac{1}{b + c}, \quad \forall i \in I_b \cup I_c$$

The chi-square similarity ρ_{χ^2} becomes $\rho_{\chi^2} = \sum_{i=1}^D \frac{2u_i v_i}{u_i + v_i} = \frac{2c}{a+b+2c}$ and hence $\frac{\rho_{\chi^2}}{2-2\rho_{\chi^2}} = \frac{c}{a+b}$.

We apply Cauchy projections, i.e., $r_i, i = 1$ to D are i.i.d. standard Cauchy.

$$\begin{aligned}
x &= \sum_{i=1}^D u_i r_i = \frac{1}{a+c} \sum_{i \in I_a} r_i + \frac{1}{a+c} \sum_{i \in I_c} r_i = \frac{a}{a+c} R_1 + \frac{c}{a+c} R_3 \\
y &= \sum_{i=1}^D v_i r_i = \frac{1}{b+c} \sum_{i \in I_b} r_i + \frac{1}{b+c} \sum_{i \in I_c} r_i = \frac{b}{b+c} R_2 + \frac{c}{b+c} R_3
\end{aligned}$$

where R_1, R_2, R_3 are i.i.d. standard Cauchy variables.

$$\begin{aligned}
& \Pr(x < 0, y < 0) \\
&= \Pr\left(\frac{a}{a+c}R_1 + \frac{c}{a+c}R_3 < 0, \frac{b}{b+c}R_2 + \frac{c}{b+c}R_3 < 0\right) \\
&= \Pr\left(R_1 < -\frac{c}{a}R_3, R_2 < -\frac{c}{b}R_3\right) \\
&= \frac{1}{\pi^2} E\left\{\left(\tan^{-1}\left(-\frac{c}{a}R_3\right) + \frac{\pi}{2}\right)\left(\tan^{-1}\left(-\frac{c}{b}R_3\right) + \frac{\pi}{2}\right)\right\}
\end{aligned}$$

By symmetry

$$\Pr(x > 0, y > 0) = \Pr(x < 0, y < 0)$$

and

$$\begin{aligned}
\Pr(\text{sign}(x) \neq \text{sign}(y)) &= 1 - \frac{2}{\pi^2} E\left\{\left(\tan^{-1}\left(-\frac{c}{a}R_3\right) + \frac{\pi}{2}\right)\left(\tan^{-1}\left(-\frac{c}{b}R_3\right) + \frac{\pi}{2}\right)\right\} \\
&= \frac{1}{2} - \frac{2}{\pi^2} E\left\{\tan^{-1}\left(\frac{c}{a}|R_3\right)\tan^{-1}\left(\frac{c}{b}|R_3\right)\right\}
\end{aligned}$$

C Proof of Lemma 6

The task is to prove the property of $Z(t_1, t_2)$, where $t_1 \geq 0, t_2 \geq 0$, and

$$Z(t_1, t_2) = -\frac{2}{\pi^2} E\left\{\tan^{-1}\left(\frac{1}{t_1}|R\right)\tan^{-1}\left(\frac{1}{t_2}|R\right)\right\} + \frac{1}{\pi} E\left\{\tan^{-1}\left(\frac{1}{t_1+t_2}|R\right)\right\}$$

Here R is a standard Cauchy random variable with density $f_R(r) = \frac{1}{\pi} \frac{1}{1+r^2}$.

At the boundaries (i.e., $t_1 = 0$, or $t_2 = 0$, or $t_1 = \infty$, or $t_2 = \infty$), we have $Z(t_1, t_2) = 0$. We hope to find other extremum points by checking the first derivatives:

$$\begin{aligned}
\frac{\partial Z(t_1, t_2)}{\partial t_1} &= \frac{2}{\pi^2} E\left\{\frac{|R|}{t_1^2 + R^2} \tan^{-1}\left(\frac{1}{t_2}|R\right)\right\} - \frac{1}{\pi} E\left\{\frac{|R|}{(t_1+t_2)^2 + R^2}\right\} \\
\frac{\partial Z(t_1, t_2)}{\partial t_2} &= \frac{2}{\pi^2} E\left\{\frac{|R|}{t_2^2 + R^2} \tan^{-1}\left(\frac{1}{t_1}|R\right)\right\} - \frac{1}{\pi} E\left\{\frac{|R|}{(t_1+t_2)^2 + R^2}\right\}
\end{aligned}$$

The integrals can be analytically calculated.

$$\begin{aligned}
E\left\{\frac{|R|}{(t_1+t_2)^2 + R^2}\right\} &= \frac{1}{\pi} \int_0^\infty \frac{2r}{(t_1+t_2)^2 + r^2} \frac{1}{1+r^2} dr = \frac{1}{\pi} \int_0^\infty \frac{1}{(t_1+t_2)^2 + r} \frac{1}{1+r} dr \\
&= \frac{1}{\pi} \frac{1}{(t_1+t_2)^2 - 1} \log \frac{t+1}{t+(t_1+t_2)^2} \Big|_0^\infty = \frac{1}{\pi} \frac{\log((t_1+t_2)^2)}{(t_1+t_2)^2 - 1}
\end{aligned}$$

Using the integral table [10, 4.535.14], we obtain

$$E\left\{\frac{|R|}{t_1^2 + R^2} \tan^{-1}\left(\frac{1}{t_2}|R\right)\right\} = \frac{1}{\pi} \int_{-\infty}^\infty \tan^{-1}\left(\frac{r}{t_2}\right) \frac{r}{r^2 + t_1^2} \frac{1}{1+r^2} dr = \frac{1}{t_1^2 - 1} \log \frac{t_1 + t_2}{1 + t_2}$$

Therefore,

$$\begin{aligned}\frac{\partial Z(t_1, t_2)}{\partial t_1} &= \frac{2}{\pi^2} \frac{1}{t_1^2 - 1} \log \frac{t_1 + t_2}{1 + t_2} - \frac{2}{\pi^2} \frac{\log(t_1 + t_2)}{(t_1 + t_2)^2 - 1} \\ \frac{\partial Z(t_1, t_2)}{\partial t_2} &= \frac{2}{\pi^2} \frac{1}{t_2^2 - 1} \log \frac{t_1 + t_2}{1 + t_1} - \frac{2}{\pi^2} \frac{\log(t_1 + t_2)}{(t_1 + t_2)^2 - 1}\end{aligned}$$

Setting $\frac{\partial Z(t_1, t_2)}{\partial t_1} = 0$ and $\frac{\partial Z(t_1, t_2)}{\partial t_2} = 0$ yields

$$\begin{aligned}\frac{1}{t_1^2 - 1} \log \frac{t_1 + t_2}{1 + t_2} - \frac{\log(t_1 + t_2)}{(t_1 + t_2)^2 - 1} &= 0 \\ \frac{1}{t_2^2 - 1} \log \frac{t_1 + t_2}{1 + t_1} - \frac{\log(t_1 + t_2)}{(t_1 + t_2)^2 - 1} &= 0\end{aligned}$$

which means we must have

$$g(t_1, t_2) = (t_1^2 - 1) \log \frac{t_1 + t_2}{1 + t_1} - (t_2^2 - 1) \log \frac{t_1 + t_2}{1 + t_2} = 0$$

which obviously holds when $t_1 = t_2$. We just need to make sure there are no other solutions with $t_1 \neq t_2$.

Note that, for fixed $t_2 < 1$, $g(t_1, t_2)$ is a concave function in t_1 ; and for fixed $t_2 > 1$, $g(t_1, t_2)$ is a convex function in t_1 . To see this fact, we compute the derivatives of $g(t_1, t_2)$:

$$\begin{aligned}\frac{\partial g(t_1, t_2)}{\partial t_1} &= 2t_1 \log \frac{t_1 + t_2}{1 + t_1} + (t_1^2 - 1) \left(\frac{1}{t_1 + t_2} - \frac{1}{1 + t_1} \right) - \frac{t_2^2 - 1}{t_1 + t_2} = 2t_1 \log \frac{t_1 + t_2}{1 + t_1} - t_2 + 1 \\ \frac{\partial^2 g(t_1, t_2)}{\partial t_1^2} &= 2 \log \frac{t_1 + t_2}{1 + t_1} + 2t_1 \left(\frac{1}{t_1 + t_2} - \frac{1}{1 + t_1} \right)\end{aligned}$$

For fixed t_2 , we have $\frac{\partial^2 g(t_1, t_2)}{\partial t_1^2} \Big|_{t_1=\infty} = 0$. We can check the third derivative:

$$\frac{\partial^3 g(t_1, t_2)}{\partial t_1^3} = 4 \left(\frac{1}{t_1 + t_2} - \frac{1}{1 + t_1} \right) + 2t_1 \left(-\frac{1}{(t_1 + t_2)^2} + \frac{1}{(1 + t_1)^2} \right) = 2 \frac{(1 - t_2)(t_1 + 2t_2 + t_1 t_2)}{(t_1 + t_2)^2 (1 + t_1)^2}$$

which means that, if $t_2 < 1$, then $\frac{\partial^3 g(t_1, t_2)}{\partial t_1^3} \geq 0$ (i.e., $\frac{\partial^2 g(t_1, t_2)}{\partial t_1^2} \leq 0$), and if $t_2 > 1$ then $\frac{\partial^3 g(t_1, t_2)}{\partial t_1^3} \leq 0$ (i.e., $\frac{\partial^2 g(t_1, t_2)}{\partial t_1^2} \geq 0$).

For fixed t_2 , the function $g(t_1, t_2)$ is either convex or concave, which means $g(t_1, t_2) = 0$ has at most two solutions. In the meanwhile, we know that if $t_1 = t_2$, or $t_1 = 1$, then $g(t_1, t_2) = 0$.

Now it remains to check the case of $t_1 = 1$. When $t_1 \rightarrow 1$, we have

$$\frac{\partial Z(t_1, t_2)}{\partial t_1} \rightarrow \frac{2}{\pi^2} \frac{1}{(t_1 + 1)(t_2 + 1)} - \frac{2}{\pi^2} \frac{\log(1 + t_2)}{(1 + t_2)^2 - 1} \neq 0$$

This way, we can conclude that the extremum points can only happen at $t_1 = t_2$ and hence we only need to study a one-dimensional function.

$$h(t) = \frac{\pi^2}{2} \frac{\partial Z(t_1, t_2)}{\partial t_1} \Big|_{t_1=t_2=t} = \frac{1}{t^2 - 1} \log \frac{2t}{1 + t} - \frac{\log(2t)}{(2t)^2 - 1} \quad (26)$$

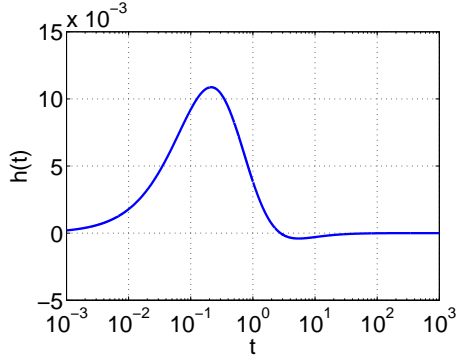


Figure 12: The function $h(t)$ as defined in (26). It is positive when $t < t^* = 2.7793457703$ and negative when $t > t^*$.

$h(t)$ is plotted in Figure 12. The solution to $h(t) = 0$ is $t^* = 2.7793457703$. Moreover, $h(t) > 0$ if $t < t^*$ and $h(t) < 0$ if $t > t^*$. This means the maximum of $Z(t_1, t_2)$ is attained at $t_1 = t_2 = t^*$.

Now it remains to compute the maximum value of $Z(t)$, where

$$\begin{aligned} Z(t) &= Z(t_1 = t, t_2 = t) \\ &= -\frac{2}{\pi^2} E \left\{ \tan^{-1} \left(\frac{1}{t} |R| \right) \tan^{-1} \left(\frac{1}{t} |R| \right) \right\} + \frac{1}{\pi} E \left\{ \tan^{-1} \left(\frac{1}{2t} |R| \right) \right\} \\ &= \int_0^\infty \left\{ -\frac{2}{\pi^2} \left(\tan^{-1} \left(\frac{r}{t} \right) \right)^2 + \frac{1}{\pi} \tan^{-1} \left(\frac{r}{2t} \right) \right\} \frac{2}{\pi} \frac{1}{1+r^2} dr \end{aligned}$$

which reaches the maximum at $t = t^*$.

References

- [1] <http://googleresearch.blogspot.com/2010/04/lessons-learned-developing-practical.html>.
- [2] Bogdan Alexe, Thomas Deselaers, and Vittorio Ferrari. What is an object? In *CVPR*, pages 73–80, San Francisco, CA, 2010.
- [3] Leon Bottou. <http://leon.bottou.org/projects/sgd>.
- [4] Olivier Chapelle, Patrick Haffner, and Vladimir N. Vapnik. Support vector machines for histogram-based image classification. *IEEE Trans. Neural Networks*, 10(5):1055–1064, 1999.
- [5] Moses S. Charikar. Similarity estimation techniques from rounding algorithms. In *STOC*, pages 380–388, Montreal, Quebec, Canada, 2002.
- [6] Rong-En Fan, Kai-Wei Chang, Cho-Jui Hsieh, Xiang-Rui Wang, and Chih-Jen Lin. Liblinear: A library for large linear classification. *Journal of Machine Learning Research*, 9:1871–1874, 2008.
- [7] Yoav Freund, Sanjoy Dasgupta, Mayank Kabra, and Nakul Verma. Learning the structure of manifolds using random projections. In *NIPS*, Vancouver, BC, Canada, 2008.
- [8] Jerome H. Friedman, F. Baskett, and L. Shustek. An algorithm for finding nearest neighbors. *IEEE Transactions on Computers*, 24:1000–1006, 1975.

- [9] Michel X. Goemans and David P. Williamson. Improved approximation algorithms for maximum cut and satisfiability problems using semidefinite programming. *Journal of ACM*, 42(6):1115–1145, 1995.
- [10] Izrail S. Gradshteyn and Iosif M. Ryzhik. *Table of Integrals, Series, and Products*. Academic Press, New York, seventh edition, 2007.
- [11] Matthias Hein and Olivier Bousquet. Hilbertian metrics and positive definite kernels on probability measures. In *AISTATS*, pages 136–143, Barbados, 2005.
- [12] Piotr Indyk. Stable distributions, pseudorandom generators, embeddings, and data stream computation. *Journal of ACM*, 53(3):307–323, 2006.
- [13] Piotr Indyk and Rajeev Motwani. Approximate nearest neighbors: Towards removing the curse of dimensionality. In *STOC*, pages 604–613, Dallas, TX, 1998.
- [14] Yugang Jiang, Chongwah Ngo, and Jun Yang. Towards optimal bag-of-features for object categorization and semantic video retrieval. In *CIVR*, pages 494–501, Amsterdam, Netherlands, 2007.
- [15] Thorsten Joachims. Training linear svms in linear time. In *KDD*, pages 217–226, Pittsburgh, PA, 2006.
- [16] Fuxin Li, Guy Lebanon, and Cristian Sminchisescu. A linear approximation to the χ^2 kernel with geometric convergence. Technical report, arXiv:1206.4074, 2013.
- [17] Ping Li. Very sparse stable random projections for dimension reduction in l_α ($0 < \alpha \leq 2$) norm. In *KDD*, San Jose, CA, 2007.
- [18] Ping Li. Estimators and tail bounds for dimension reduction in l_α ($0 < \alpha \leq 2$) using stable random projections. In *SODA*, pages 10 – 19, San Francisco, CA, 2008.
- [19] Ping Li, Trevor J. Hastie, and Kenneth W. Church. Improving random projections using marginal information. In *COLT*, pages 635–649, Pittsburgh, PA, 2006.
- [20] Ping Li and Arnd Christian König. b-bit minwise hashing. In *WWW*, pages 671–680, Raleigh, NC, 2010.
- [21] Ping Li, Art B Owen, and Cun-Hui Zhang. One permutation hashing. In *NIPS*, Lake Tahoe, NV, 2012.
- [22] Ping Li and Cun-Hui Zhang. A new algorithm for compressed counting with applications in shannon entropy estimation in dynamic data. In *COLT*, 2011.
- [23] S. Muthukrishnan. Data streams: Algorithms and applications. *Foundations and Trends in Theoretical Computer Science*, 1:117–236, 2 2005.
- [24] Noam Nisan. Pseudorandom generators for space-bounded computations. In *Proceedings of the twenty-second annual ACM symposium on Theory of computing*, STOC, pages 204–212, 1990.
- [25] Gennady Samorodnitsky and Murad S. Taqqu. *Stable Non-Gaussian Random Processes*. Chapman & Hall, New York, 1994.
- [26] Bernhard Schölkopf and Alexander J. Smola. *Learning with Kernels*. The MIT Press, Cambridge, MA, 2002.
- [27] Shai Shalev-Shwartz, Yoram Singer, and Nathan Srebro. Pegasos: Primal estimated sub-gradient solver for svm. In *ICML*, pages 807–814, Corvallis, Oregon, 2007.

- [28] Andrea Vedaldi and Andrew Zisserman. Efficient additive kernels via explicit feature maps. *IEEE Trans. Pattern Anal. Mach. Intell.*, 34(3):480–492, 2012.
- [29] Sreekanth Vempati, Andrea Vedaldi, Andrew Zisserman, and C. V. Jawahar. Generalized rbf feature maps for efficient detection. In *BMVC*, pages 1–11, Aberystwyth, UK, 2010.
- [30] Gang Wang, Derek Hoiem, and David A. Forsyth. Building text features for object image classification. In *CVPR*, pages 1367–1374, Miami, Florida, 2009.
- [31] Jinjun Wang, Jianchao Yang, Kai Yu, Fengjun Lv, Thomas S. Huang, and Yihong Gong. Locality-constrained linear coding for image classification. In *CVPR*, pages 3360–3367, San Francisco, CA, 2010.
- [32] Kilian Weinberger, Anirban Dasgupta, John Langford, Alex Smola, and Josh Attenberg. Feature hashing for large scale multitask learning. In *ICML*, pages 1113–1120, 2009.
- [33] Qiang Yang and Xindong Wu. 10 challenge problems in data mining research. *International Journal of Information Technology and Decision Making*, 5(4):597–604, 2006.
- [34] Haiquan (Chuck) Zhao, Nan Hua, Ashwin Lall, Ping Li, Jia Wang, and Jun Xu. Towards a universal sketch for origin-destination network measurements. In *NPC*, 2011.