A Heterogeneous High Dimensional Approximate Nearest Neighbor Algorithm

Moshe Dubiner

ACKNOWLEDGMENT

I would like to thank Phil Long and David Pablo Cohn for reviewing rough drafts of this paper and suggesting many clarifications. The remaining obscurity is my fault.

M. Dubiner is with Google, e-mail: moshe@google.com

Manuscript submitted to IEEE Transactions on Information Theory on March 3 ,2007; revised February 20, 2009.

A Heterogeneous High Dimensional Approximate Nearest Neighbor Algorithm

Abstract

We consider the problem of finding high dimensional approximate nearest neighbors. Suppose there are d independent rare features, each having its own independent statistics. A point x will have $x_i = 0$ denote the absence of feature i, and $x_i = 1$ its existence. Let $p_{i,jk}$ be the probability that $x_i = j$ and $y_i = k$ for "near" points x, y versus the random $(p_{i,00} + p_{i,01})(p_{i,10} + p_{i,11})$ for random pairs. Sparsity means that usually $x_i = 0 : p_{01} + p_{10} + p_{11} = o(1)$. Distance between points is a variant of the Hamming distance. Dimensional reduction converts the sparse heterogeneous problem into a lower dimensional full homogeneous problem. However we will see that the converted problem can be much harder to solve than the original problem. Instead we suggest a direct approach, which works in the dense case too. It consists of T tries. In try t we rearrange the coordinates in increasing order of $\lambda_{t,i}$ which satisfy $\frac{p_{i,00}}{(1-r_{t,i})(p_{i,00}+p_{i,01})^{\lambda_{t,i}}+r_{t,i}} + \frac{p_{i,11}}{(1-r_{t,i})(p_{i,10}+p_{i,11})^{\lambda_{t,i}}+r_{t,i}} = 1$ where $0 < r_{t,i} < 1$ are uniform random numbers, and the $p's$ are the coordinates' statistical parameters. The points are lexicographically ordered, and each is compared to its neighbors in that order.

We analyze generalizations of this algorithm, show that it is optimal in some class of algorithms, and estimate the necessary number of tries to success. It is governed by an information like function, which we call small leaves bucketing forest information. Any doubts whether it is "information" are dispelled by another paper, where bucketing information is defined.

I. Introduction

This paper is motivated by the following practical problem. We have a large (billions) corpora of documents, some in English and some in French (for example). Some (say 1 percent) of the French documents are translations of some of the English documents (or vice versa). How can these parallel pairs be found? A natural approach is to use an English-French dictionary of words (or phrases). Let the dictionary contain d parallel word pairs: English word E_i is paired with the French word F_i for $1 \leq i \leq d$. Hence each text generates a a binary vector of d bits, where the i'th coordinate indicates whether E_i (or F_i) appears in the text:

$$z = (z_1, z_2, \ldots, z_d) \qquad z_i = 0, 1 \tag{1}$$

The dictionary is much larger than most texts, so most of the bits will be 0 (sparsity). We are not counting the number of word appearances, nor their locations. Of course not all parallel word pairs are created equal: some are reliable translations of each other, while other are not. Some words are rare, while others are common. These important differences are expressed by the probability matrix of the i'th words pair

$$P_i = \begin{pmatrix} p_{i,00} & p_{i,01} \\ p_{i,10} & p_{i,11} \end{pmatrix} \qquad 1 \leq i \leq d \tag{2}$$

$$p_{i,00} + p_{i,01} + p_{i,10} + p_{i,11} = 1 \tag{3}$$

where $p_{i,00}$ is the probability that for a random translation text pair neither the English text contains word E_i nor the French text contains word F_i. Similarly $p_{i,01}$ is the probability that the English text does not contain E_i but the French text contains F_i etc. These probabilities are estimated from a large number (millions) of pairs. We make the theoretical simplification of considering the word pairs independent: the probability of having the paired English-French vectors x,y is

$$\mathrm{Prob}_{\mathrm{pair}}(x, y) = \prod_{i=1}^{d} p_{i,x_i\, y_i} \tag{4}$$

The probability of having random English-French vectors x,y is

$$\mathrm{Prob}_{\mathrm{rand}}(x, y) = \prod_{i=1}^{d} p_{i,x_i *} p_{i,* y_i} \tag{5}$$

where * means "don't care", so $p_{i,j*}, p_{i,*k}$ are the marginal probabilities

$$p_{i,j*} = p_{i,j0} + p_{i,j1} \tag{6}$$

$$p_{i,*k} = p_{i,0k} + p_{i,1k} \tag{7}$$

Of course these assumptions do not hold in practice, but they provide a relevant fully defined mathematical problem: assuming n_0 English and n_1 French vectors are generated according to (4),(5), find the parallel pairs.

The bichromatic (English-French) setting is not essential, and all our results are valid for the monochromatic case, where of course $p_{i,jk} = p_{i,kj}$. However it seems that bichromatism adds clarity. This is reminiscent of "fast" matrix multiplication: even if one cares only for square matrices, rectangular matrix multiplication is relevant.

Information theory limits what can be achieved without a bound on the number of operations. Consider for simplicity the case where there is a single nonrandom vector pair. Than we can not pin down that pair into less than W possibilities where

$$\ln W = \ln n_0 + \ln n_1 - \sum_{i=1}^{d} I(P_i) \tag{8}$$

and the mutual information in coordinate i is

$$I(P_i) = \sum_{j,k=0,1} p_{i,jk} \ln \frac{p_{i,jk}}{p_{i,j*}\, p_{i,*k}} \tag{9}$$

When $\ln W < 0$ there is hope. But can one do better than comparing all $n_0 n_1$ pairs?

As far as we are aware this is the first attempt to adopt the information theory probabilistic setting to a vector pairs finding problem. However it similar to the very well known problem of finding approximate nearest neighbors. The simplest connection is for

$$P_i = \begin{pmatrix} p_i/2 & (1-p_i)/2 \\ (1-p_i)/2 & p_i/2 \end{pmatrix} \tag{10}$$

In that case

$$\mathrm{Prob}_{\mathrm{rand}}(x, y) = 4^{-d} \tag{11}$$

$$\mathrm{Prob}_{\mathrm{pair}}(x, y) = \alpha\, \mathrm{e}^{-\mathrm{dist}(x,y)} \tag{12}$$

where α is a constant and dist is an L_1 metric

$$\mathrm{dist}(x, y) = \sum_{i=1}^{d} (x_i \neq y_i)\ \ln \frac{p_i}{1-p_i} \tag{13}$$

The parallel pairs are exactly the dist close pairs. There is very extensive theoretical literature dealing with finding them. In low dimensions (d fixed, $n_0, n_1 \to \infty$) it can be done in $\mathrm{O}(n_0 + n_1) \ln(n_0 + n_1)$ elementary operations or even faster, depending on details. However in our case the dimension d is at least of order $\ln(n_0 + n_1)$. The large dimensional problem is addressed by the Locality Sensitive Hashing theory of Indyk and Motwani[8]. Their starting abstraction is to drop any data details besides the metric distances and space type (here L_1). It makes sense as far as evaluating pairing suggestions is concerned. However there is no reason to assume that a good parallel pairs finding algorithm will use only metric information. In fact we will see in section VII that the both the LSH bounds and the random projection algorithms are not suited to sparse problems.

Why do we use a probabilistic setting? The homogeneous $P_i = \begin{pmatrix} p_{00} & p_{01} \\ p_{10} & p_{11} \end{pmatrix}$ problem for $1 \leq i \leq d$ is similar to deterministically require that each parallel vector pair has $p_{00}(1+o(1))d$ common 0's, $p_{01}(1+o(1))d$ English 0's versus French 1's etc, and each nonparallel vector pair has $p_{0*}p_{*0}(1+o(1))d$ common 0's etc. However the heterogeneous (i dependent) probabilistic problem has no nice deterministic counterpart. That is a reason why the standard information theory setup is probabilistic. We accept that standard for the approximate nearest neighbor problem too.

There exist numerous practical algorithms for finding approximate nearest neighbors, early examples are Manber [7], Broder [3] and Gennaro, Savino and Zezula [6]. Broder arranges the d coordinates in a uniformly random order, and computes for each vector a signature, consisting of the location of its first nonzero coordinate. Let us call the set of all vectors with some fixed signature a **bucket**. English-French pairs falling into the same bucket are examined for parallelism. The probability that a parallel pair falls into the same bucket is much larger than for a non parallel pair. It is still small, so one makes T bucketing tries, each using a separate random permutation.

II. Main Results

Definition 2.1: For this paper, the standard data model is the following. Let the sets $X_0, X_1 \subset \{0,1,...,b-1\}^d$ of cardinalities $n_0 = \#X_0,\ n_1 = \#X_1$. The $n_0 n_1$ $X_0 \times X_1$ vector pairs are divided into 3 classes: parallel, random and monochromatic. For a parallel pair the probability that both their i'th coordinates equal j is $p_{i,jj}$ with independence between coordinates. Denote the probability of disagreement by

$$p_{i,bb} = 1 - \sum_{j=0}^{b-1} p_{i,jj} \tag{14}$$

We assume that each vector can be a member of at most one parallel pair. For a random pair the probability that both their i'th coordinates equal j is $p_{i,j*}p_{i,*j}$ with independence between coordinates. Monochromatic pairs are allowed only when $p_{i,j*} = p_{i,*j}$, for the technical purpose of duplicating a monochromatic set of points $X_0 = X_1 = X$. The probability that both their i'th coordinates equal j is $p_{i,j*}$ with independence between coordinates.

Definition 2.2: A **bucketing tree** algorithm computes a signature for all vectors $z \in X_0 \cup X_1 \subset \{0,1,\ldots,b-1\}^d$. Recursively the signature is either

- Empty (we are at a tree leaf, which is called a bucket).
- The value z_i of some coordinate i, followed by the signature of some bucketing tree.
- A random integer in some range, followed by the signature of some bucketing tree.

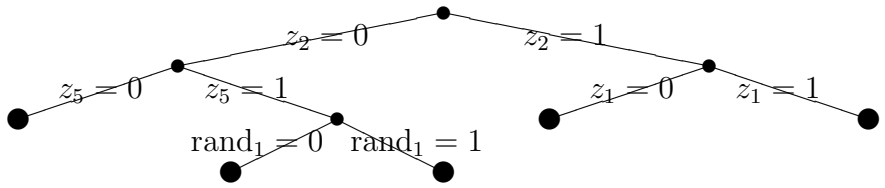
The bucketing tree's work is

$$W = \max(n_0, n_1, n_0 n_1 Q) \tag{15}$$

where the **random collision number** Q is the expected number of falls into the same bucket for a random vectors pair. The bucketing tree's success probability S is the probability that a random parallel vectors pair falls into the same bucket (have the same signature).

A **bucketing forest** algorithm is the union of T bucketing tree algorithms. Its work W is the sum of the tree's works. Its success probability S is the probability that a random parallel vectors pair falls into the same bucket for at least one tree.

In the work definition n_0 is the number of X_0 signature computations, n_1 is the number of X_1 signature computations and $n_0 n_1 Q$ is the expected number of random putative parallel pairs. For example the bucketing tree



has

$$Q = p_{2,0*}p_{2,*0}(p_{5,0*}p_{5,*0} + p_{5,1*}p_{5,*1}/2) + p_{2,1*}p_{2,*1}(p_{1,0*}p_{1,*0} + p_{1,1*}p_{1,*1}) \tag{16}$$

$$S = p_{2,00}(p_{5,00} + p_{5,11}/2) + p_{2,11}(p_{1,00} + p_{1,11}) \tag{17}$$

The success probability of a bucketing forest is complicated, because every tree's failure decreases the success probability of later trees. Nevertheless we will prove in section A

Theorem 2.1: For any bucketing forest

$$\ln W \geq \max_{\substack{0 \leq \lambda \leq 1 \\ 1 \leq \mu}} \left[\ln \max(n_0, n_1) + \lambda \ln \min(n_0, n_1) + \mu \ln S - \sum_{i=1}^{d} G(P_i, \lambda, \mu) \right] \tag{18}$$

where $G(P, \lambda, \mu)$ is some function jointly convex in λ, μ, defined in (123). Moreover the bound is asymptotically tight in the following sense. For any $\epsilon > 0$ there exits a constant $a(\epsilon)$ such that for any dimension $d \geq 1$, probabilities satisfying $p_{i,jj} = 0$ or $p_{i,jj} > \epsilon$ for any $1 \leq i \leq d,\ 0 \leq j < b$

and parameters $n_0, n_1,\ S \leq 1 \leq W$ satisfying the above relation there exits a bucketing forest with work $\tilde{W} \leq W\mathrm{e}^{a(\epsilon)+\epsilon d}$ and success $\tilde{S} \geq S\mathrm{e}^{-a(\epsilon)-\epsilon d}$.

It is a nice result, but bucketing trees rely too heavily on the probability estimates being correct. A less risky approach is

Definition 2.3: The **maximal leaf probability** of a bucketing forest is the maximum over the leaves of the probability that a random X_0 vector falls into it. A small leaves **bucketing forest** is a bucketing forest whose maximal leaf probability is at most

$$1/\min(n_0, n_1) \tag{19}$$

The small leaves condition insures that the work equals

$$W = \max(n_0, n_1)T \tag{20}$$

and it is easy to enforce that the number of X_0 vectors in a bucket is of order $\max(1, n_0/n_1)$. A slight modification of the proof of theorem 2.1 gives

Theorem 2.2: For any small leaves bucketing forest

$$\ln W \geq \max_{\substack{0\leq\lambda\leq 1\\ 1\leq\mu}} \left[\ln\max(n_0, n_1) + \lambda \ln\min(n_0, n_1) + \mu \ln S - \sum_{i=1}^{d} F(P_i, \lambda, \mu)\right] \tag{21}$$

where $F(P, \lambda, \mu)$ is some function jointly convex in λ, μ, defined in (126). Moreover the bound is asymptotically tight as in theorem 2.1.

Of course $F(P_i, \lambda, \mu) \leq G(P_i, \lambda, \mu)$.

What is the meaning formulas (21)? Denote the critical (optimizing) parameters by λ_c, μ_c. Intuitively

- If we double $\min(n_0, n_1)$ without increasing the work, the success probability is approximately reduced by factor 2^{μ_c}.
- If we double $\min(n_0, n_1)$, the work needed to achieve the same success probability as before is approximately multiplied by 2^{λ_c}.
- If we delete coordinate i, the work needed to achieve the same success probability as before is on average multiplied by $e^{F(P_i,\lambda_c,\mu_c)}$. In particular $F(P_i, \lambda_c, \mu_c) = 0$ means that under current conditions coordinate i is useless.

The striking similarity between (21) and the information bound (8) motivates us to call $F(P_i, \lambda, \mu)$ the **small leaves bucketing forest information** function. A devil's advocate might argue

that we are making much out of coincidence. The true probabilistic nearest neighbors information should be defined by allowing general algorithms, not just bucketing forests, and counting the necessary number of operations till most parallel pairs are listed. General algorithms are beyond our power, but in [5] we consider the very large class of fixed bucketing algorithms, and prove a similar formula for its performance, involving the unmistakenly mutual information like **bucketing information**.

The proofs of theorem 2.2 is constructive, but it is a cumbersome construction. For large success probability (for instance $S > 1/2$) we will prove that the following algorithm is asymptotically optimal.

Definition 2.4: The **random bucketing forest** algorithm generates a small leaves bucketing tree as follows. Let $0 < r_1, r_2, \ldots, r_d < 1$ be independent uniform $[0,1]$ random real numbers, one per coordinate. For each coordinate i let its **random exponent** $0 \leq \lambda_i \leq 1$ be the unique solution of the monotone equation

$$\sum_{j=0}^{b-1} \frac{p_{i,jj}}{(1-r_i)p_{i,j*}^{\lambda_i} + r_i} = 1 \tag{22}$$

When there is no solution in range the coordinate is ignored, so we are left with $\tilde{d} \leq d$ coordinates. The random exponents define a random subpermutation π by $\lambda_{\pi_1} < \lambda_{\pi_2} < \cdots < \lambda_{\pi_{\tilde{d}}}$. Each vector $z \in X_0 \cup X_1$ generates a signature $(z_{\pi_1}, z_{\pi_2}, \ldots, z_{\pi_m})$ where m is the minimal integer such that

$$\prod_{i=1}^{m} p_{\pi_i, z_{\pi_i}*} \leq 1/\min(n_0, n_1) \tag{23}$$

If there is no such m we generate signature $(z_{\pi_1}, z_{\pi_2}, \ldots, z_{\pi_{\tilde{d}}}, k)$ with k a uniformly random integer in range $0 \leq k < \min(n_0, n_1) \prod_{i=1}^{\tilde{d}} p_{\pi_i, z_{\pi_i}*}$

The extra k signature is similar to adding many $\begin{pmatrix} 1/4 & 1/4 \\ 1/4 & 1/4 \end{pmatrix}$ junk coordinates to the data. The number of trees T for success is asymptotically

$$\ln T \sim \max_{0 \leq \lambda \leq 1} \left[\lambda \ln \min(n_0, n_1) - \sum_{i=1}^{d} F(P_i, \lambda, \infty) \right] \tag{24}$$

In the sparse case $b = 2,\ p_{i,01} + p_{i,10} + p_{i,11} = o(1)$ we want only small λ_i . Then the bound

$$\frac{p_{i,11}}{(1-r_i)p_{i,1*}^{\lambda_i} + r_i} = 1 - \frac{p_{i,00}}{(1-r_i)p_{i,0*}^{\lambda_i} + r_i} \leq 1 - p_{i,00} \tag{25}$$

$$p_{i,1*}^{\lambda_i} \geq 1 - \frac{1}{(1-r_i)\left(1+\frac{p_{i,11}}{p_{i,01}+p_{i,10}}\right)} \tag{26}$$

is close to equality, and we need not solve an implicit equation.

A nice variant is

Definition 2.5: The **random bucketing permutations** algorithm is similar to the random bucketing forest algorithm up to the random exponents generation stage. Then we lexicographically sort all the $n_0 + n_1$ points, with lower exponent coordinates given precedence over larger exponent coordinates, and the coordinate values $0, 1, \ldots, b-1$ arbitrarily arranged, even without consistency. Ties are randomly broken. Each X_1 point is putatively paired with the preceding and following $10 \max(1, n_0/n_1)$ X_0 points.

The bucketing permutations algorithm has two nice properties

- For sparse data, we can consider only nonzero value coordinates.
- During each try at most $20 \max(n_0, n_1)$ pairs are compared, even when our probabilities are wrong.

Notice that when the random bucketing forest algorithm succeeds, the corresponding random bucketing permutations algorithm succeeds with probability at least 0.9 (because the probability that the bucket size is at least ten times its expected size is at most 1/10).

There must be a reason for such a relatively nice optimal algorithm. What we have are only proofs.

III. The Homogeneous Marginally Bernoulli(1/2) Example

The homogeneous marginally Bernoulli(1/2) example is easy to analyze. The analysis is non-generalizable, but the issues remain. Let

$$P_i = \begin{pmatrix} p/2 & (1-p)/2 \\ (1-p)/2 & p/2 \end{pmatrix} \tag{27}$$

For some $p > 1/2$. Without restricting generality let $n_0 \leq n_1$. We randomly choose $m \approx \log_2 n_0$ out of the d coordinates, and let their values define the buckets. This is a random algorithm solving a random problem, so we have two levels of randomness. Usually when we will compute probabilities or expectations, it will be with respect to these two sources together. The expected number of vector pairs falling into the same buckets is $n_0 n_1 2^{-m}$, while the probability that a parallel pair falls into the same bucket (success) is p^m. (These statements are true assuming only

model randomness). If the dimension d is large enough so that each try uses different coordinates, the success probability of T tries is $1-(1-p^m)^T$. Hence taking $T \approx p^{-m}/\ln 2 \approx n_0^{\log_2 1/p}/\ln 2$ results in a success probability of about $1/2$. The expected number of vector comparisons is of order

$$W = O\left(n_0^{\log_2 1/p} n_1\right) \tag{28}$$

When coordinates are shared between tries things are more complicated. Let us consider one parallel pair. The probability that they agree on k bits is $\begin{pmatrix} d \\ k \end{pmatrix} p^k (1-p)^{d-k}$, in particular $k > pd$ with probability nearly $1/2$. The probability of success in a single try conditioned on k is $\begin{pmatrix} k \\ m \end{pmatrix} \Big/ \begin{pmatrix} d \\ m \end{pmatrix}$. Hence we get a decent success probability of about $(1-1/\mathrm{e})/2$ by taking

$$T \approx \begin{pmatrix} d \\ m \end{pmatrix} \Big/ \begin{pmatrix} pd \\ m \end{pmatrix} = \prod_{i=0}^{m-1} \frac{1-i/d}{p-i/d} \tag{29}$$

The total number of signatures is $T(n_0+n_1)$, and the total expected number of comparisons is

$$W = n_0 2^{-m} T n_1 \approx n_0 n_1 \prod_{i=0}^{m-1} \frac{1-i/d}{2(p-i/d)} \tag{30}$$

Hence increasing m above $(2p-1)d$ is counterproductive, and the best choice is

$$m \approx \min[\log_2 n_0, (2p-1)d] \tag{31}$$

Another observation is that $d >> \ln(n_0)$ suffices to make the effects of tries' interdependence asymptotically negligible.

IV. The Marginally Bernoulli(1/2) Example

Let us see what our theory says about $n_0 \le n_1$

$$P_i = \begin{pmatrix} p_i/2 & (1-p_i)/2 \\ (1-p_i)/2 & p_i/2 \end{pmatrix} \qquad 1 \le i \le d \tag{32}$$

Equation (22) is

$$2\frac{p_i/2}{(1-r_i)2^{-\lambda_i} + r_i} = 1 \tag{33}$$

which can be recast as

$$2^{-\lambda_i} = \frac{p_i - r_i}{1-r_i} \tag{34}$$

The small leaves bucketing forest function will turn out to be in this case

$$F(P_i,\lambda,\infty)=G(P_i,\lambda,\infty)=\begin{cases} p_i\ln\frac{p_i}{2^{-\lambda}}+(1-p_i)\ln\frac{1-p_i}{1-2^{-\lambda}} & p_i\geq 2^{-\lambda} \\ 0 & p_i\leq 2^{-\lambda}\end{cases} \tag{35}$$

The critical exponent attains (24). The extremal condition is

$$\sum_{i=1}^{d}\max\left[\frac{p_i-2^{-\lambda_c}}{1-2^{-\lambda_c}},0\right]=\log_2 n_0 \tag{36}$$

For the homogeneous case $p_1=p_2=\cdots=p_d=p$

$$2^{-\lambda_c}=\frac{pd-\log_2 n_0}{d-\log_2 n_0} \tag{37}$$

Notice that $\log_2 n_0<(2p-1)d$ is equivalent to $\lambda_c<1$.

An extreme heterogeneous example is $n_0=n_1=10^9$, 50 coordinates have $p_i=0.9$, and other 950 coordinates have $p_i=0.7$. Solving (36) results in $2^{-\lambda_c}\doteq 0.75$, hence the relatively bad coordinates are thrown out and we need about $n_0^{\lambda_c}\mathrm{e}^{-\sum_{i=1}^{d}F(P_i,\lambda_c)}\doteq 144$ tries. If we chose 30 out of the 1000 coordinates uniformly, few would have had 0.9 probability, and we would have needed about 20000 tries. In general the probability that coordinate 1 will have larger random exponent than coordinate 2 when $p_1>p_2$ is

$$\frac{1}{2}\frac{1-p_1}{1-p_2} \tag{38}$$

Hence the probability that a 0.7 coordinate precedes a 0.9 coordinate is 0.17 . However the chance that a 0.7 coordinate will be ranked among the first 30 is very small.

V. Intuition for the Marginally Bernoulli(1/2) Example

In this section we will present an intuitive argument for the previous section's formulas, without any rigor. Let us order the coordinates in decreasing order of importance

$$p_1\geq p_2\geq\cdots\geq p_d \tag{39}$$

Moreover let us bunch coordinates together into g groups of $d_1,d_2,\ldots,d_g$ coordinates, where $\sum_{h=1}^{g}d_h=d$, and the members of group h all have the same probability q_h

$$p_{d_1+\cdots+d_{h-1}+1}=\cdots=p_{d_1+\cdots+d_h}=q_h \tag{40}$$

Out of the d_h coordinates in group h, a parallel pair will agree in approximately $q_h d_h$ 'good' coordinates. Let us make things simple by pretending that this is the exact value (never mind

that it is not an integer). We want to choose $m \approx \log_2 n_0$ coordinates and compare pairs which agree on them. The greedy approach seems to choose as many as possible from the group 1, but conditional greed disagrees. Let us pick the first coordinate randomly from group 1. If it is bad, the whole try is lost. If it is good, group 1 is reduced to size $d_1 - 1$, out of which $q_1 d_1 - 1$ are good. Hence the probability that a remaining coordinate is good is reduced to $\frac{q_1 d_1 - 1}{d_1 - 1}$. After taking k coordinates out of group 1, its probability decreases to $\frac{q_1 d_1 - k}{d_1 - k}$. Hence after taking $k = \frac{q_1 - q_2}{1 - q_2} d_1$ coordinates, group 1 merges with group 2. We will randomly chose coordinates from this merged group till its probability drops to q_3. At that point the probability of a second group coordinate to be chosen is $\frac{q_2 - q_3}{1 - q_3}$ while the probability of a first group coordinate being picked either before or after the union is

$$\frac{q_1 - q_2}{1 - q_2} + \left(1 - \frac{q_1 - q_2}{1 - q_2}\right) \frac{q_2 - q_3}{1 - q_3} = \frac{q_1 - q_3}{1 - q_3} \tag{41}$$

This goes on till at some $q_l = p_c$ we have m coordinates. Then the probability that coordinate i is chosen is

$$\max\left[\frac{p_i - p_c}{1 - p_c}, 0\right] \tag{42}$$

The cutoff probability is determined by

$$\sum_{i=1}^{d} \max\left[\frac{p_i - p_c}{1 - p_c}, 0\right] \approx m \tag{43}$$

The previous equation can be iteratively solved. However it is better to look from a different angle. For each try we will have to generate d independent uniform $[0, 1]$ random real numbers

$$0 < r_1, r_2, \ldots, r_d < 1 \tag{44}$$

one random number per coordinate. Then we take coordinate i iff

$$r_i \leq \frac{p_i - p_c}{1 - p_c} \tag{45}$$

Let us reverse direction. Generate r_i first, and then compute for which p_c's coordinate i is taken:

$$p_c \leq 2^{-\lambda_i} = \max\left[\frac{p_i - r_i}{1 - r_i}, 0\right] \tag{46}$$

We call λ_i the random exponent of coordinate i (random because it is r_i dependent). Remember that $p_c > 0$ so $\lambda_i = \infty$ means that for that value of r_i coordinate i can not be used. Now which value of p_c will get us m coordinates? There is no need to solve equations. Sort the λ_i's in nondecreasing order, and pick out the first m. Hence $p_c = 2^{-\lambda_c}$ where the cutoff exponent λ_c is the value of the $m'th$ ordered random exponent.

VI. An Unlimited Homogeneous Data Example

Suppose we have an unlimited amount of data $d \to \infty$ of the same type

$$P_i = \begin{pmatrix} p_{00} & p_{01} \\ p_{10} & p_{11} \end{pmatrix} \tag{47}$$

What is the success probability of a small leaves bucketing tree? Our initial estimate was the following. The probability that a parallel pair will agree in a single coordinate is $p_{00} + p_{11}$. The amount of information in a single English coordinate is $-p_{0*} \ln p_{0*} - p_{1*} \ln p_{1*}$ so we will need about $m \approx \frac{\ln n_0}{-p_{0*} \ln p_{0*} - p_{1*} \ln p_{1*}}$ coordinates, and the success probability is estimated by $(p_{00} + p_{11})^m \approx n_0^{-\frac{\ln(p_{00}+p_{11})}{p_{0*} \ln p_{0*} + p_{1*} \ln p_{1*}}}$ This estimate turns out to be disastrously wrong. For the bad matrix $\begin{pmatrix} 1-2\rho & \rho \\ \rho & 0 \end{pmatrix}$ with small ρ it suggests exponent $1/\ln(1/\rho)$, while clearly it should be 1. The interested reader might pause to figure out what went wrong, and how this argument can be salvaged.

There is an almost exact simple answer with a surprising flavor.

Theorem 6.1: The success probability of S a small leaves bucketing tree for unlimited $\begin{pmatrix} p_{00} & p_{01} \\ p_{10} & p_{11} \end{pmatrix}$ data and $n = \min(n_0, n_1)$ is within bounds

$$(n/\min(p_{0*}, p_{1*}))^{-\lambda} \leq S \leq n^{-\lambda} \tag{48}$$

where $0 \leq \lambda < 1$ is the unique solution of

$$p_{00} p_{0*}^{-\lambda} + p_{11} p_{1*}^{-\lambda} = 1 \tag{49}$$

or $\lambda = 1$ when there is no solution in range.

Proof: Let $S(n)$ denote the success probability of a single tree, with leaf sizes at most $1/n$ for any real $n > 0$. Clearly $S(n) = 1$ for $n \leq 1$. When $\lambda < 1 < n$ some coordinate is used, so recursively

$$S(n) = p_{00} S(n p_{0*}) + p_{11} S(n p_{1*}) \tag{50}$$

and the theorem follows by induction. When $\lambda = 1$ no coordinates will be chosen ($\tilde{d} = 0$) so $S = 1/\lceil n \rceil$. ■

VII. The Downside of Random Projections

The Indyk-Motwani paper [8] introduces a metric based, worst case analysis. In general no average upper bound can replace a worst case upper bound, and the reverse holds for lower bounds. Still some comparison is unavoidable. The Indyk-Motwani theory is (nonessentially) monochromatic $X_0 = X_1 = X \subset \mathrm{R}^d$, $n_0 = n_1 = n$. There is a metric dist and constants $c \geq 1,\ r > 0$ such that parallel pairs $x, y \in X$ satisfy

$$\mathrm{dist}(x, y) \leq r \tag{51}$$

while random pairs satisfy

$$\mathrm{dist}(x, y) \geq cr \tag{52}$$

They have shown that for an L_1 metric $\mathrm{dist}(x, y) = \sum_{i=1}^{d} b_i |x_i - y_i|$ there exists an algorithm (called LSH) with success probability $S = 1 - o(1)$ and work $W = O(n^{1+1/c})$.

Let us consider uniform symmetric

$$P_i = \begin{pmatrix} p_{00} & p_{01} \\ P_{10} & p_{11} \end{pmatrix} \qquad p_{01} = p_{10} \tag{53}$$

The only L_1 metric that makes sense here is the Humming distance. The Chernof inequality implies that for any $0 < \epsilon \to 0$ there exists $\delta > 0$ such that for a parallel pair (x, y) $\mathrm{Prob}\{|\mathrm{dist}(x, y) - 2p_{01}d| > \epsilon d\} < \mathrm{e}^{-\delta d}$ and for a random pair $\mathrm{Prob}\{|\mathrm{dist}(x, y) - 2p_{0*}p_{1*}d| > \epsilon d\} < \mathrm{e}^{-\delta d}$. Taking $d = \lceil 3(\ln n)/\delta \rceil$ insures that with probability at least $1 - 1/n$ all n^2 distances are within the ϵ bounds, so

$$|c - p_{0*}p_{1*}/p_{01}| < O(\epsilon) = o(1) \tag{54}$$

The familiar

$$P_i = \begin{pmatrix} p/2 & (1-p)/2 \\ (1-p)/2 & p/2 \end{pmatrix} \tag{55}$$

has $c = 1/(2 - 2p) + o(1)$. The resulting bound

$$W = O(n^{3-2p+o(1)}) \tag{56}$$

is more pessimistic than the real performance of random coordinate selection (28) $W = O(n^{\log_2 2/p+o(1)})$.

Let us now consider a typical sparse bits matrix: for a small ρ

$$P_i = \begin{pmatrix} 1 - 3\rho & \rho \\ \rho & \rho \end{pmatrix} \tag{57}$$

In the previous section we saw that small leaves bucketing forest requires $W = O(n^{1+\lambda})$ comparisons where $(1-3\rho)(1-2\rho)^{-\lambda}+\rho(2\rho)^{-\lambda} = 1$. Bounding $(1-2\rho)^{-\lambda} > 1$ gives $\lambda < \frac{\ln 3}{\ln 1/2\rho}$

$$W = O\left(n^{1+\frac{\ln 3}{\ln 1/2\rho}}\right) \tag{58}$$

In particular the exponent tends to 1 as ρ tends to 0. The LSH ratio is $c = 2(1-2\rho)+o(1)$, resulting in

$$W = O(n^{1+\frac{1}{2-4\rho}+o(1)}) \tag{59}$$

As far as the LSH theory is concerned $\begin{pmatrix} 1-3\rho & \rho \\ \rho & \rho \end{pmatrix}$ and $\begin{pmatrix} 1/2-1/(8-16\rho) & 1/(8-16\rho) \\ 1/(8-16\rho) & 1/2-1/(8-16\rho) \end{pmatrix}$ generate equaly difficult nearest nieghbor problems with $c = 2(1-2\rho)+o(1)$. If one accepts that, dimensional reduction by random projection can look good.

Datar Indyk Immorlica and Mirrokni [4] recommend a specific random projections approach that "works well on data that is extremely high-dimensional but sparse". Their optimal choice $r \to \infty$ results in a binary hash function $h(x) = \operatorname{sign}\left(\sum_{i=1}^{d} z_i C_i\right)$ where $(z_1, z_2, \ldots, z_d) \in R^d$ is any vector and $C_1, C_2, \ldots, C_d$ are independent Cauchy random variables (density $\frac{1}{\pi(1+x^2)}$). Both ± 1 values have probability $1/2$, so one has to concatenate $m \approx \log_2 n_0$ binary hash functions in order to determine a bucket. Now consider two parallel vectors. They will have approximately ρd 1's in common, and each will have approximately ρd 1's where the other has zeroes. The sum of ρd independent Cauchy random variables has the same distribution as ρd times a single Cauchy random variable, so the probability that the two parallel vectors get the same hash bit is approximately

$$\operatorname{Prob}\{\operatorname{sign}(C_1 + C_2) = \operatorname{sign}(C_1 + C_3)\} = 2/3 \tag{60}$$

Hence the amount of actual work is even worse than (59):

$$W = O((3/2 + o(1))^m n) = O(n^{\log_2 3 + o(1)}) \tag{61}$$

Random projections allow efficient approximate computation of distances between vectors. But they make finding parallel vector pairs harder than before the data was scrambled. We encourage the interested reader to look at his favorite dimensional reduction scheme, and see that the $\ln 1/2\rho$ factor is really lost.

VIII. A Small Leaves Bucketing Forest Bound

In this section we will prove a lower bound on the performance of small leaves bucketing forest algorithms. The proof is elegant, tricky and hard to generalize.

Definition 8.1:

$$F(P,\lambda,\infty)=\max_{0\le r\le 1}\left[\ln(1-r)+\sum_{j=0}^{b-1}p_{jj}\ln\left(1-r+rp_{j*}^{-\lambda}\right)\right] \tag{62}$$

Other representations are

$$F(P,\lambda,\infty)=\max_{\substack{0\le u_0,u_1,\ldots,u_b\\ u_0+u_1+\ldots+u_b=1}}\left[\lambda\sum_{j=0}^{b-1}u_j\ln\frac{1}{p_{j*}}+u_b\ln u_b+(1-u_b)\ln(1-u_b)+\right. \tag{63}$$

$$\left.+\sum_{j=0}^{b-1}\left(p_{jj}\ln p_{jj}-u_j\ln u_j-(p_{jj}-u_j)\ln(p_{jj}-u_j)\right)\right] \tag{64}$$

$$F(P,\lambda,\infty)=\min_{\substack{0\le q_0,\ldots,q_b\\ \sum_{j=0}^{b}q_j=1\\ \sum_{j=0}^{b-1}q_jp_{j*}^{-\lambda}\le 1}}\sum_{j=0}^{b}p_j\ln\frac{p_j}{q_j} \tag{65}$$

Equivalence follows from evaluation at the extremal arguments. When $\sum_{j=0}^{b-1}p_{jj}p_{j*}^{-\lambda}\le 1$ the extremal $r=0$ and of course $F(P,\lambda,\infty)=0$. Otherwise the extremal r satisfies

$$\sum_{j=0}^{b-1}\frac{p_{jj}}{(1-r)p_{j*}^{\lambda}+r}=1 \tag{66}$$

The extremal u's are $u_j=\frac{rp_{jj}}{(1-r)p_{j*}^{\lambda}+r}$ for $0\le j<b$ and $u_b=1-r$. The extremal q's are $q_j=\frac{p_{jj}}{1-r+rp_{j*}^{-\lambda}}$ for $0\le j<b$ and $q_b=\frac{p_{bb}}{1-r}$.

Theorem 8.1: The success probability S of a small leaves bucketing tree whose leaves all have probabilities at most $1/n$ is at most

$$S\le n^{-\lambda}\prod_{i=1}^{d}\max\left(1,\sum_{j=0}^{b-1}p_{i,jj}p_{i,j*}^{-\lambda}\right) \tag{67}$$

for any $0\le\lambda\le 1$. We do not assume $p_{i,jj}\le p_{i,j*}$.

Proof: Use induction. The base is trivial: condition $1\le 1/n$ implies $1\le n^{-\lambda}$ (this is where $0\le\lambda$ is used). Without losing generality the tree starts with coordinate 1

$$S(n)\le\sum_{j=0}^{b-1}p_{1,jj}S(np_{1,j*})\le\sum_{j=0}^{b-1}p_{1,jj}(np_{1,j*})^{-\lambda}\prod_{i=2}^{d}\max\left(1,\sum_{j=0}^{b-1}p_{i,jj}p_{i,j*}^{-\lambda}\right) \tag{68}$$

or for a random split into k possibilities (this is where $\lambda \leq 1$ is used)

$$S(n) \leq k^{-1} S(n/k) \leq k^{-1} (n/k)^{-\lambda} \prod_{i=1}^{d} \max \left(1, \sum_{j=0}^{b-1} p_{i,jj} p_{i,j*}^{-\lambda} \right) \tag{69}$$

The maximization with 1 is necessary because coordinates can be ignored. ■

Theorem 8.2: Suppose a small leaves bucketing forest has leaves probability at most $1/n$, contains T trees and its success probability is S. Then for any $0 \leq \lambda \leq 1$

$$\ln T \geq \lambda \ln N + \ln S/2 - \sum_{i=1}^{d} F(P_i, \lambda, \infty) - \sqrt{\frac{4}{S} \sum_{i=1}^{d} V(P_i, \lambda)} \tag{70}$$

where

$$V(P_i, \lambda) = \sum_{j=0}^{b} p_{i,j} \left(\ln \frac{p_{i,j}}{q_{i,j}} - \sum_{k=0}^{b} p_{i,k} \ln \frac{p_{i,k}}{q_{i,k}} \right)^2 \tag{71}$$

and the $q_{i,j}$'s are the minimizing arguments from F's definition (65)

Proof: The previous theorem provides a good bound for the success probability of a single tree, but it is not tight for a forest, because of dependence: the failure of each tree increases the failure probability of other trees. Now comes an interesting argument. The success probability of a bucketing forest (or even more general bucketing schemes) can be written as follows. Consider a parallel pair. Let $y_i = j$ denote both having value j in coordinate i (probability $p_{i,jj}$), and $y_i = b$ denote disagreement in coordinate i (probability $p_{i,bb}$). Let $0 \leq A(y) \leq 1$ denote the probability that $y \in \{0, \ldots, b\}^d$ is accepted by the forest (it is 0 or 1 when there is no random branching). Clearly

$$S = \mathrm{E}[A] = \sum_{y \in \{0,\ldots,b\}^d} A(y) \prod_{i=1}^{d} p_{i,y_i y_i} \tag{72}$$

What happens when we replace $p_{i,jj}$ by $q_{i,j} \geq 0$ for all i, j in the success formula?

$$S(Q) = \sum_{y \in \{0,\ldots,b\}^d} A(y) \prod_{i=1}^{d} q_{i,y_i} = E\left(A \mathrm{e}^{-Z} \right) \tag{73}$$

where

$$Z(y) = \sum_{i=1}^{d} \ln \frac{p_{i,y_i y_i}}{q_{i,y_i}} \tag{74}$$

For any scalar z

$$S = \mathrm{E}((Z \geq z)A) + \mathrm{E}\left((Z < z) \mathrm{e}^{Z} A \mathrm{e}^{-Z} \right) \leq \mathrm{Prob}\{Z \geq z\} + e^z S(Q) \tag{75}$$

We insist upon $\sum_{j=0}^{b} q_{i,j} = 1$ so that we can use the previous lemma to bound $S(Q)$. Notice that $q_{i,j} \leq p_{i,j*}$ is not required.

$$S(Q) \leq Tn^{-\lambda} \prod_{i=1}^{d} \max \left(1, \sum_{j=0}^{b-1} q_{i,j} p_{i,j*}^{-\lambda} \right) \tag{76}$$

The other term is handled by the Chebyshev bound: for $z > \mathrm{E}(Z)$

$$\mathrm{Prob}\{Z \geq z\} \leq \frac{\mathrm{Var}(Z)}{(z - \mathrm{E}(Z))^2} \tag{77}$$

Together

$$S \leq \frac{\mathrm{Var}(Z)}{(z - \mathrm{E}(Z))^2} + e^z S(Q) \tag{78}$$

The reasonable choice of

$$z = \mathrm{E}(Z) + \sqrt{2\mathrm{Var}(Z)/S} \tag{79}$$

results in

$$S \leq 2e^{\mathrm{E}(Z) + \sqrt{2\mathrm{Var}(Z)/S}} S(Q) \tag{80}$$

■

Notice that this proof gives no indication that the bound is tight, nor guidance towards constructing an actual bucketing forest, (except for telling which coordinates to throw away).

IX. Conclusion

To sum up, we present three things:

1) A practical approximate nearest neighbor algorithm 2.5.
2) An information style performance estimate 2.2.
3) A warning against dimensional reduction of sparse data in section VII.

References

[1] N. Alon Private Communication.

[2] A. Andoni, P. Indyk *Near-Optimal Hashing Algorithms for Approximate Nearest Neighbor in High Dimensions* FOCS 2006.

[3] A. Broder. *Identifying and Filtering Near-Duplicate Documents* Proc. FUN, 1998.

[4] M. Datar, P. Indyk, N. Immorlica and V. Mirrokni *Locality-Sensitive Hashing Scheme Based on p-Stable Distributions* Proc. Sympos. on Computational Geometry, 2004.

[5] Moshe Dubiner *Bucketing Coding and Information theory for the Statistical High Dimensional Nearest Neighbor Problem* To be Published.

[6] C.Gennaro, P.Savino and P.Zezula *Similarity Search in Metric Databases through Hashing* Proc. ACM workshop on multimedia, 2001.

[7] U. Manber *Finding similar files in a large file system* Proceedings of the USENIX Winter 1994 Technical Conference

[8] P. Indyk and R. Motwani. *Approximate Nearest Neighbor: Towards Removing the Curse of Dimensionality* Proc. 30th Annu. ACM Sympos. Theory Comput., 1998.

[9] R.M. Karp, O. Waarts, and G. Zweig. *The Bit Vector Intersection Problem* Proc. 36th Annu. IEEE Sympos. Foundations of Computer Science, pp. 621-630, 1995.

APPENDIX A
PROOF OF THEOREMS 2.1,2.2

The proof we are about to give is elaborate but constructive. It systematically computes the bucketing forest information function. Bucketing trees are recursively defined, but we failed to inductively bound their performance. Instead we use tensor products.

Definition A.1: Suppose we have a bucketing forest defined over $\{0,\ldots,b-1\}^d$ having T trees, maximal leaf probability $1/N$, collision number Q and success probability S. Suppose another forest defined over $\{0,\ldots,b-1\}^{\tilde{d}}$ has $\tilde{T}$ trees etc. Their **tensor product** is the forest containing T_1T_2 trees, one for each pair. They are generated by taking a tree from the first forest and grafting the same tree from the second forest on each leaf. In particular the tensor product will have $T\tilde{T}$ trees, maximal leaf probability $1/(N\tilde{N})$, collision number $Q\tilde{Q}$ and success probability $S\tilde{S}$.

Theorem A.1: For any bucketing forest with T trees, work per point Q and success probability S there exist numbers $0 \le \alpha,\beta$, $0 \le u_{i,j} \le v_{i,j}$ and $0 \le v_{i,b} \le u_{i,b}$ for $1 \le i \le d,\ 0 \le j < b$ such that $\sum_{j=0}^{b} u_{i,j} = \sum_{j=0}^{b} v_{i,j} = 1$ and

$$\ln T \ge Z_4 - Z_3 - \alpha \ge 0 \tag{81}$$

$$\ln Q \ge Z_4 - Z_2 - \alpha - \beta \tag{82}$$

$$\ln S \le -Z_1 - \alpha - \beta \tag{83}$$

where

$$Z_1 = \sum_{i=1}^{d}\sum_{j=0}^{b} v_{i,j} \ln \frac{v_{i,j}}{p_{i,jj}} \tag{84}$$

$$Z_2 = \sum_{i=1}^{d}\sum_{j=0}^{b-1} u_{i,j} \ln \frac{1}{p_{i,j*}p_{i,*j}} \tag{85}$$

$$Z_3 = \sum_{i=1}^{d} \left((1-u_{i,b}) \ln(1-u_{i,b}) - \sum_{j=0}^{b-1} u_{i,j} \ln u_{i,j} \right) \tag{86}$$

$$Z_4 = \sum_{i=1}^{d} \left(-u_{i,b} \ln u_{i,b} + \sum_{j=0}^{b-1} ((v_{i,j}-u_{i,j}) \ln(v_{i,j}-u_{i,j}) - v_{i,j} \ln v_{i,j}) \right) \tag{87}$$

On the other hand for any parameters satisfying the above conditions and $\epsilon > 0$ there exists a constant $a(e)$ such that for any $m \geq 1$ there exits a bucketing forest defined on md coordinates, with the original probabilities each repeated m times, such that the forest has at most $T^m \mathrm{e}^{a(\epsilon)+\epsilon m}$ trees, at most $Q^m \mathrm{e}^{a(\epsilon)+\epsilon m}$ collision number and at least $S^m \mathrm{e}^{-a(\epsilon)-\epsilon m}$ success probability.

Proof: The inverse direction of constructing a bucketing forest according to parameters is easy. We need only do it for $d = 1$ and then take a tensor product. Let $m \to \infty$, $\mathbf{u}_j = \lfloor m u_{1,j} \rfloor$, $\mathbf{v}_j = \lfloor m v_{1,j} \rfloor$ for $0 \leq j < b$ and $\mathbf{u}_b = m - \sum_{j=0}^{b-1} \mathbf{u}_j$, $\mathbf{v}_b = m - \sum_{j=0}^{b-1} \mathbf{v}_j$. We construct a uniform forest of random

$$T_m = \lceil \mathrm{e}^{m(Z_4 - Z_3 - \alpha)} \rceil \tag{88}$$

trees, each involving randomly chosen $m - \mathbf{u}_{1,b}$ coordinates out of the m available. The signature is the ordered values of these coordinates, followed by a random number in range 1 to $\lceil \mathrm{e}^{m\beta} \rceil$. However only leaves that attain value j $\mathbf{u}_j$ times are kept (the rest can get an extra very large random signature). Hence each tree contains $L_m = K_m \lceil \mathrm{e}^{m\beta} \rceil$ leaves, where

$$K_m = \begin{pmatrix} m - \mathbf{u}_b \\ \mathbf{u}_0, \ldots, \mathbf{u}_{b-1} \end{pmatrix} = \mathrm{e}^{m(Z_3 + o(1))} \tag{89}$$

The collision number is

$$Q_m = T_m K_m \prod_{j=0}^{b-1} (p_{1,j*} p_{1,*j})^{\mathbf{u}_j} / \lceil \mathrm{e}^{m\beta} \rceil = \mathrm{e}^{m(Z_4 - Z_2 - \alpha - \beta + o(1))} \tag{90}$$

The success probability (averaged over the random choices of trees) is

$$S_m = 1 - \sum_{\substack{0 \leq \nu_1, \ldots, \nu_b \\ \nu_0 + \cdots + \nu_b = m}} m! \prod_{j=0}^{b} \frac{p_{1,jj}^{\nu_j}}{\nu_j!} \left(1 - \prod_{j=0}^{b-1} \begin{pmatrix} \nu_j \\ \mathbf{u}_j \end{pmatrix} \Big/ \begin{pmatrix} m \\ m - \mathbf{u}_b \end{pmatrix} \right)^{T_m} \tag{91}$$

$$S_m \geq m! \prod_{j=0}^{b} \frac{p_{1,jj}^{\mathbf{v}_j}}{\mathbf{v}_j!} \Big/ \left(1 + \frac{1}{T_m} \begin{pmatrix} m \\ m - \mathbf{u}_b \end{pmatrix} \Big/ \prod_{j=0}^{b-1} \begin{pmatrix} \mathbf{v}_j \\ \mathbf{u}_j \end{pmatrix} \right) \tag{92}$$

$$m! \prod_{j=0}^{b} \frac{p_{1,jj}^{\mathbf{v}_j}}{\mathbf{v}_j!} = \mathrm{e}^{-m(Z_1 + o(1))} \tag{93}$$

$$\left(\begin{array}{c} m \\ m-\mathbf{u}_b \end{array} \right) \Big/ \prod_{j=0}^{b-1} \left(\begin{array}{c} \mathbf{v}_j \\ \mathbf{u}_j \end{array} \right) = \mathrm{e}^{m(Z_4 - Z_3 + o(1))} \tag{94}$$

Suppose we are given a bucketing forest depending upon d coordinates and extra $\tilde{d}$ random branching vertices (at which a random signature is computed). Let us raise the forest to tensor power m. A subforest of the tensor forest will be called uniform iff each leaf involves $0 \le \mathbf{u}_{i,j}$ branches where coordinate i and its tensor copies have value j, and $0 \le \tilde{\mathbf{u}}_{\tilde{i}} \le m$ random branches that are the tensor copies of random branch $\tilde{i}$. Denote $\mathbf{u}_{i,b} = m - \sum_{j=0}^{b-1} \mathbf{u}_{i,j} \ge 0$. The tensor power forest is split into at most $(m+1)^{bd+\tilde{d}}$ uniform subforests (each leaf goes into one uniform subforest). We take the subforest with the largest success probability, and denote $\mathbf{u}_{i,j} = mu_{i,j}$. We will let $m \to \infty$, and there will be a subsequence of m's whose u's converge. The uniform subforest has success probability S_m, T_m trees, and collision number Q_m . Clearly

$$T \ge T_m^{1/m}, \quad Q \ge Q_m^{1/m}, \quad S \le (m+1)^{(bd+\tilde{d})/m} S_m^{1/m} \tag{95}$$

Notice that $(m+1)^{(bd+\tilde{d})/m} = 1 + o(1)$.

The $\tilde{\mathbf{u}}_1 + \cdots + \tilde{\mathbf{u}}_{\tilde{d}}$ random branchings will simply reduce the success probability and the collision number by the same factor $\mathrm{e}^{-m\beta}$ ($\beta \ge 0$), so let us count reduced leaves i.e. leaves without them. How many reduced leaves K_m can a single uniform tree have? Without restricting generality let it start with coordinate 1. Then

$$K_m(\mathbf{u}_{1,0}, \ldots, \mathbf{u}_{1,b-1}, \ldots) \le K_m(\mathbf{u}_{1,0} - 1, \ldots, \mathbf{u}_{1,b-1}, \ldots) + \cdots + K_m(\mathbf{u}_{1,0}, \ldots, \mathbf{u}_{1,b-1} - 1, \ldots) \tag{96}$$

so by induction

$$K_m \le \prod_{i=1}^{d} \frac{(m - \mathbf{u}_{i,b})!}{\prod_{j=0}^{b-1} \mathbf{u}_{i,j}!} \tag{97}$$

$$\frac{1}{m} \ln K_m \le Z_3 + o(1) \tag{98}$$

The total number of reduced leaves L_m satisfies

$$L_m \le T_m K_m \tag{99}$$

Denoting

$$\alpha \equiv Z_4 - \limsup \frac{1}{m} \ln L_m \tag{100}$$

we have proven most of (81)

$$\ln T \geq \frac{1}{m} \ln T_m \geq Z_4 - Z_3 - \alpha \tag{101}$$

Each reduced leaf has collision number e^{-mZ_2} so we get (82)

$$Q_m = \mathrm{e}^{-m(Z_2+\beta)} L_m \tag{102}$$

$$\ln Q \geq Z_4 - Z_2 - \alpha - \beta \tag{103}$$

For a random md dimensional parallel pair, let $0 \leq \mathbf{v}_{i,j} \leq m$ (for $1 \leq i \leq d$, $0 \leq j < b$) denote the number of times coordinate type i attains value j for both vectors, and $\mathbf{v}_{i,b} = m - \sum_{j=0}^{b-1} \mathbf{v}_{i,j}$ is the number of mistakes. The $\mathbf{v}$'s can have at most $(m+1)^{bd}$ values, each contributing something to the success probability of our reduced uniform forest. Let $\mathbf{v}_{i,j} = mv_{i,j}$'s contribute the most. Again there is a subsequence of m's for which the v's converge. Then

$$S_m \leq (m+1)^{bd} \prod_{i=1}^{d} \left(m! \prod_{j=0}^{b} \frac{p_{i,jj}^{\mathbf{v}_{i,j}}}{\mathbf{v}_{i,j}!} \right) \mathrm{e}^{-m\beta} \tag{104}$$

$$\ln S \leq -Z_1 - \beta \tag{105}$$

Assuming the $\mathbf{v}$'s, the success probability of a single leaf is

$$s_m = \prod_{i=1}^{d} \frac{\prod_{j=0}^{b-1} [\mathbf{v}_{i,j}(\mathbf{v}_{i,j}-1)\cdots(\mathbf{v}_{i,j}-\mathbf{u}_{i,j}+1)]}{m(m-1)\cdots(\mathbf{u}_{i,b}+1)} \tag{106}$$

$$\frac{1}{m} \ln s_m = -Z_4 + o(1) \tag{107}$$

Summing up over the leaves we get (83)

$$S_m \leq (m+1)^{bd} \prod_{i=1}^{d} \left(m! \prod_{j=0}^{b} \frac{p_{i,jj}^{\mathbf{v}_{i,j}}}{\mathbf{v}_{i,j}!} \right) \mathrm{e}^{-m\beta} L_m s_m \tag{108}$$

$$\ln S \leq -Z_1 - \alpha - \beta \tag{109}$$

We are still missing the limitations on α. Negative α can be replaced with $\alpha = 0$ because of (105). When $\alpha > Z_4 - Z_3$ we can increase β by $\alpha - Z_4 + Z_3$, and decrease α by the same amount. Inequality (101) remains valid because $\ln T \geq 0$. ■

Definition A.2: For any bucketing tree and n_0, n_1 let the point number be $n = \min(n_0, n_1)$ and the work per point be $V = W/\max(n_0, n_1) = \max(T, nQ)$, for any real $n_0, n_1 \geq 0$. Denote

$$E(P_1, \ldots, P_d) = \left\{ \frac{1}{m} \left(\ln n^m, -\ln \tilde{S_m}, \ln \tilde{V_m} \right) \;\middle|\; 0 \leq \tilde{S_m} \leq S_m,\ \tilde{V_m} \geq V_m \right\}^c \tag{110}$$

where the set depend on all md dimensional bucketing forests, with each P_i duplicated $m \geq 1$ times, and c means closure.

Lemma A.2: $E(P_1, \ldots, P_d)$ is a convex subset of R^3. It equals

$$\left\{ \frac{1}{m} \left(\ln \frac{T_m}{Q_m}, -\ln S_m, \ln T_m \right) \right\}^c + \mathrm{ConvCone}\left(\{(-1,0,0),(0,1,0),(1,0,1)\} \right) \tag{111}$$

where T_m, Q_m, S_m go over the number of trees, collision number and success probability of all md dimensional bucketing forests, with each P_i duplicated $m \geq 1$ times. The convex cone is $\{(\gamma_3 - \gamma_1, \gamma_2, \gamma_3) \mid 0 \leq \gamma_1, \gamma_2, \gamma_3\}$.

Proof: The convexity holds because the tensor product of k copies of a forest with T_m trees and $\tilde{k}$ copies of a forest with $\tilde{T}_m$ trees has $T_m^k \tilde{T}_m^{\tilde{k}}$ trees, and all other relevant quantities behave in the same way. The trivial forest (one tree, single leaf) shows that E contains the convex cone. For $n^m \geq T_m/Q_m$ $V_m = n^m Q_m$ and we can write

$$(\ln n^m, -\ln S_m, \ln V_m) = (\ln T_m/Q_m, -\ln S_m, \ln T_m) + (1,0,1)\ln(n^m Q_m/T_m) \tag{112}$$

while for $n^m \leq T_m/Q_m$ $V_m = T_m$ so

$$(\ln n^m, -\ln S_m, \ln V_m) = (\ln T_m/Q_m, -\ln S_m, \ln T_m) + (1,0,0)\ln(T_m/n^m Q_m) \tag{113}$$

■

We now recognize theorem A.1 to state

Corollary A.3:

$$E = D \cap \{(x,y,z) \mid z \geq 0\} \tag{114}$$

$$D = E + \mathrm{ConvCone}\left(\{ (0,1,-1) \} \right) = \tag{115}$$

$$= \{(Z_2 - Z_3, Z_1, Z_4 - Z_3)\} + \mathrm{ConvCone}\left((-1,0,0),(1,0,1),(0,1,-1) \right) \tag{116}$$

where the $P_i, u_{i,j}, v_{i,j}$ dependence is understood. Moreover

$$E(P_1, \ldots, P_d) = E(P_1) + \cdots + E(P_d) \tag{117}$$

Notice that $(0,1,0)$ dropped out because it is spanned by the others.

The set D is convex and contain the origin, so it is the dual of its dual

$$D = \{(x,y,z) \mid \forall\ (\alpha,\beta,\gamma) \in D^\perp \quad \alpha x - \beta y - \gamma z \leq 1\} \tag{118}$$

$$\tilde{D}^\perp = \{(\alpha,\beta,\gamma) \mid \forall\ (x,y,z) \in \tilde{D} \quad \alpha x - \beta y - \gamma z \leq 1\} \tag{119}$$

The convex cone establishes $0 \le \alpha \le \gamma \le \beta$. When $\gamma = 0$ it follows that $0 = \alpha \le \beta$ and the resulting condition $-\beta y \le 1$ follows from $y \ge 0$. When $\gamma > 0$ we can divide by it $\lambda = \alpha/\gamma, \mu = \beta/\gamma$ so

$$D = \{(x, y, z) \mid y, z \ge 0,\ \forall\ 0 \le \lambda \le 1 \le \mu\ z \ge \lambda x - \mu y - G(\lambda, \mu)\} \tag{120}$$

where

$$G(\lambda, \mu) = \max\left[\lambda(Z_2 - Z_3) - \mu Z_1 - (Z_4 - Z_3)\right] \tag{121}$$

In Full

Definition A.3: The bucketing forest information function $G(P, \lambda, \mu)$ is

$$G(P, \lambda, \mu) = \max_{\substack{0 \le u_0 \le v_0, \ldots, 0 \le u_{b-1} \le v_{b-1}, 0 \le u_b, v_b \\ u_0 + u_1 + \ldots + u_b = v_0 + v_1 + \ldots + v_b = 1}} \left[\lambda \sum_{j=0}^{b-1} u_j \ln \frac{u_j}{(1-u_b) p_{j*} p_{*j}} - \mu \sum_{j=0}^{b} v_j \ln \frac{v_j}{p_{jj}} + \right. \tag{122}$$

$$\left. + u_b \ln u_b + (1-u_b)\ln(1-u_b) + \sum_{j=0}^{b-1} \left(v_j \ln v_j - u_j \ln u_j - (v_j - u_j)\ln(v_j - u_j)\right)\right] \tag{123}$$

We have shown that when condition (18) is met, there are near optimal bucketing trees on $m \to \infty$ copies of $P_1, \ldots, P_d$. What does it imply for a single copy? For $\delta > 0$ $0 \le j < b$ let $P(\delta)$ be defined by $p_{jj}(\delta) = \mathrm{e}^{-\lceil -(\ln p_{jj})/\delta \rceil \delta}$ $0 \le j < b$ and similar actions for p_{j*} and p_{*j}. The work per point and success are changed by at most $\mathrm{e}^{-\delta d} \le V(\delta)/V \le 1$, $\mathrm{e}^{-\delta d} \le S(\delta)/S \le 1$. Suppose that $(\ln n, -\ln S, \ln V) \in E(P_1, \ldots, P_d)$. It implies $(\ln n, -\ln S + \delta d, \ln V) \in E(P_1(\delta), \ldots, P_d(\delta)) = E(P_1(\delta)) + \cdots + E(P_d(\delta))$. Requiring $p_{i,jj} > \epsilon$ or $p_{i,jj} = 0$ for all $1 \le i \le d,\ 0 \le j < b$ bounds the number of possible $P(\delta)$ by $(-(\ln \epsilon)/\delta + 1)^{3b}$. For each $P(\delta)$ apply the constructive direction of theorem A.1. Then replace $P_1(\delta), \ldots, P_d(\delta)$ by $P_1, \ldots, P_d$, increasing the work again by a factor. Taking small enough $\delta = \epsilon/2$ completes the proof of theorem 2.1.

The small leaves bucketing forest case is very similar to general bucketing forest, only a bit simpler. In stead of the collision number Q_m we have the small leaves inequality $Z_5 + \beta \ge \ln n$ where

$$Z_5 = \sum_{i=1}^{d} \sum_{j=0}^{b-1} u_{i,j} \ln \frac{1}{p_{i,j*}} \tag{124}$$

Definition A.4: The small leaves bucketing forest information function $F(P, \lambda, \mu)$ is

$$F(P, \lambda, \mu) = \max_{\substack{0 \le u_0 \le v_0, \ldots, 0 \le u_{b-1} \le v_{b-1}, 0 \le u_b, v_b \\ u_0 + u_1 + \ldots + u_b = v_0 + v_1 + \ldots + v_b = 1}} \left[\lambda \sum_{j=0}^{b-1} u_j \ln \frac{1}{p_{j*}} - \mu \sum_{j=0}^{b} v_j \ln \frac{v_j}{p_{jj}} + \right. \tag{125}$$

$$\left. + u_b \ln u_b + (1-u_b)\ln(1-u_b) + \sum_{j=0}^{b-1} \left(v_j \ln v_j - u_j \ln u_j - (v_j - u_j)\ln(v_j - u_j)\right)\right] \tag{126}$$

where u_b, v_b, p_{bb} are defined as before.

APPENDIX B
PERFORMANCE OF THE RANDOM BUCKETING FOREST ALGORITHM

It is relatively easy to prove that our algorithm satisfies the inverse direction of theorem 2.2 for $S > 1/2$. It is harder to give a good "hard" bound.

Theorem B.1: Denote $n = \min(n_0, n_1)$. Let $\epsilon > 0$ be some small parameter, and let $\lambda, r_1, r_2, \ldots, r_d$ attain

$$\min_{\lambda \geq 0} \max_{0 \leq r_1, \ldots, r_d \leq 1} \left[-(1+\epsilon)\lambda \ln n + \sum_{i=1}^{d} \sum_{j=0}^{b} p_{i,j} \left(1 - r_i + (j \neq b) r_i p_{i,j*}^{-\lambda}\right) \right] \tag{127}$$

The extrema conditions are

$$\sum_{i=1}^{d} \sum_{j=0}^{b-1} p_{i,j} \frac{-r_i \ln p_{i,j*}}{(1-r_i)p_{i,j*}^{\lambda} + r_i} = (1+\epsilon) \ln n \tag{128}$$

and $r_i = 0$ or

$$\sum_{j=0}^{b-1} \frac{p_{i,j}}{(1-r_i)p_{i,j*}^{\lambda} + r_i} = 1 \qquad 1 \leq i \leq d \tag{129}$$

Suppose that for some $\delta < 1/5$

$$\sum_{i=1}^{d} \sum_{j=0}^{b} p_{i,j} \Bigg(\ln[1 - r_i + (j \neq b) r_i p_{i,j*}^{-\lambda}] - \tag{130}$$

$$- \sum_{k=0}^{b} p_{i,k} \ln[1 - r_i + (k \neq b) r_i p_{i,k*}^{-\lambda}] \Bigg)^2 \leq \epsilon^2 \delta \lambda^2 (\ln n)^2 \tag{131}$$

$$\sum_{i=1}^{d} \sum_{j=0}^{b-1} p_{i,j} \left(\frac{-r_i \ln p_{i,j*}}{(1-r_i)p_{i,j*}^{\lambda} + r_i} - \sum_{k=0}^{b-1} p_{i,k} \frac{-r_i \ln p_{i,k*}}{(1-r_i)p_{i,k*}^{\lambda} + r_i} \right)^2 \leq \epsilon^2 \delta (\ln n)^2 / 4 \tag{132}$$

$$\sum_{i=1}^{d} \sum_{j=0}^{b-1} p_{i,j} \frac{r_i(1-r_i)[\ln p_{i,j*}]^2}{\left[(1-r_i)p_{i,j*}^{\lambda} + r_i\right]^2} \leq \epsilon^2 \delta (\ln n)^2 / 8 \tag{133}$$

Then the random bucketing forest algorithm with T trees where

$$\ln T \geq \ln \frac{1}{\delta} + (1+3\epsilon)\lambda \ln n - \sum_{i=1}^{d} \sum_{j=0}^{b} p_{i,j} \left(1 - r_i + (j \neq b) r_i p_{i,j*}^{-\lambda}\right) \tag{134}$$

has success probability

$$S \geq 1 - 5\delta \tag{135}$$

The alarmingly complicated small variance conditions are asymptotically valid, because both the variances and $\ln n$ grow linearly with the dimension d. However there is no guarantee that

they can be always met. Indeed the upper bound is of the Chernof inequality large deviation type, and can be a poor estimate in some cases.

Definition B.1: Let Y, Z be joint random variables. We denote by Y_Z the conditional type random variable Y with its probability density multiplied by $\frac{e^Z}{\mathrm{E}[e^Z]}$. In the discrete case Z, Y would have values y_i, z_i with probability p_i. Then Y_Z has values y_i with probability $\frac{p_i e^{z_i}}{\sum_j p_j e^{z_j}}$.

Lemma B.2: For any random variable Z, and $\lambda \geq 0$

$$\ln \mathrm{Prob}\left\{Z \geq \mathrm{E}\left[Z_{\lambda Z}\right]\right\} \leq \ln \mathrm{E}\left[e^{\lambda Z}\right] - \lambda \mathrm{E}\left[Z_{\lambda Z}\right] \tag{136}$$

$$\ln \mathrm{Prob}\left\{Z \geq \mathrm{E}\left[Z_{\lambda Z}\right] - \sqrt{2\mathrm{Var}\left[Z_{\lambda Z}\right]}\right\} \geq \tag{137}$$

$$\geq \ln \mathrm{E}\left[e^{\lambda Z}\right] - \lambda \mathrm{E}\left[Z_{\lambda Z}\right] - \ln 2 - \lambda\sqrt{2\mathrm{Var}\left[Z_{\lambda Z}\right]} \tag{138}$$

Proof: The upper bound is the Chernof bound. The lower bound combines the Chebyshev inequality

$$\mathrm{Prob}\left\{|Z_{\lambda Z} - \mathrm{E}[Z_{\lambda Z}]| \leq \sqrt{2\mathrm{Var}[Z_{\lambda Z}]}\right\} \geq \frac{1}{2} \tag{139}$$

with the fact that the condition in the curly bracket bounds the densities ratio:

$$\ln \frac{e^{\lambda Z}}{\mathrm{E}\left[e^{\lambda Z}\right]} = \ln \frac{e^{\lambda Z_{\lambda Z}}}{\mathrm{E}\left[e^{\lambda Z}\right]} \leq -\ln \mathrm{E}\left[e^{\lambda Z}\right] + \lambda \mathrm{E}\left[Z_{\lambda Z}\right] + \lambda\sqrt{2\mathrm{Var}\left[Z_{\lambda Z}\right]} \tag{140}$$

■

It is amusing, and sometimes useful to note that

$$\mathrm{E}[Z_{\lambda Z}] = \frac{\partial \ln \mathrm{E}\left[e^{\lambda Z}\right]}{\partial \lambda} \tag{141}$$

$$\mathrm{Var}[Z_{\lambda Z}] = \frac{\partial^2 \ln \mathrm{E}[e^{\lambda Z}]}{\partial \lambda^2} \tag{142}$$

We will now prove the theorem B.1.

Proof: Let $\lambda \geq 0$ be a parameter to be optimized. Let $w \in \{0,1\}^d$ be the random Bernoulli vector

$$w_i = (\lambda_i \leq \lambda) \tag{143}$$

where λ_i is the i'th random exponent. In a slight abuse of notation let $0 \leq r_i \leq 1$ denote not a random variable but a probability

$$r_i = \mathrm{Prob}\{w_i == 1\} = \mathrm{Prob}\{\lambda_i \leq \lambda\} \tag{144}$$

We could not resist doing that because equation (129) is still valid under this interpretation. Another point of view is to forget (129) and consider r_i a parameter to be optimized. Again let $y \in \{0, 1, \ldots, b\}^d$ denote the state of a parallel pair. Let us consider a single tree, conditioned on both y and w. When

$$Z(y, w) = \sum_{i=1}^{d} \ln \left[1 - w_i + (y_i \neq b) w_i p_{i,y_i*}^{-1}\right] \geq \ln n \tag{145}$$

is satisfied, the coordinates i for which $w_i = 1$ are "good" ($y_i \neq b$) and the product of their marginal probabilities $\prod_{\substack{1 \leq i \leq d \\ w_i = 1}} p_{i,y_i*} \leq 1/n$ so no further coordinates will consulted. Summing over w gives success probability of a single tree, conditioned over y to be at least

$$S(y) \geq \sum_{w \in \{0,1\}^d} \prod_{i=1}^{d} [(1 - w_i)(1 - r_i) + w_i r_i][Z(y, w) \geq \ln n]$$

In short

$$S(y) \geq \operatorname{Prob}\{Z(y) \geq \ln N\} \tag{146}$$

Conditioning over y makes the successes of random trees independent of each other, hence the conditional success probability T tries is at least

$$S_T(y) \geq 1 - (1 - S(y))^T \geq \frac{TS(y)}{1 + TS(y)} \tag{147}$$

Averaging over y bounds the success probability S of the algorithm by

$$S \geq \sum_{y \in \{0,1,\ldots,b\}^d} \prod_{i=1}^{d} p_{i,y_i y_i} \cdot \left[\frac{TS(y)}{1 + TS(y)}\right] \tag{148}$$

In short

$$S \geq \mathrm{E}\left[\frac{TS(y)}{1 + TS(y)}\right] \tag{149}$$

Now we must get our hands dirty. The reverse Chernof inequality is

$$\ln \operatorname{Prob}\left\{Z(y) \geq \mathrm{E}\left[Z(y)_{\lambda Z(y)}\right] - \sqrt{2\mathrm{Var}\left[Z(y)_{\lambda Z(y)}\right]}\right\} \geq \tag{150}$$

$$\geq \ln \mathrm{E}\left[e^{\lambda Z(y)}\right] - \lambda \mathrm{E}\left[Z(y)_{\lambda Z(y)}\right] - \ln 2 - \lambda\sqrt{2\mathrm{Var}\left[Z(y)_{\lambda Z(y)}\right]} \tag{151}$$

Denoting

$$U(y) = \ln \mathrm{E}\left[e^{\lambda Z(y)}\right] = \sum_{i=1}^{d} \ln[1 - r_i + (y_i \neq b) r_i p_{i,y_i*}^{-\lambda}] \tag{152}$$

$$V(y) = \frac{\partial U(y)}{\partial \lambda} = \mathrm{E}\left[Z(y)_{\lambda Z(y)}\right] = \sum_{\substack{1 \leq i \leq d \\ y_i \neq b}} \frac{-r_i \ln p_{i,y_i*}}{(1 - r_i) p_{i,y_i*}^{\lambda} + r_i} \tag{153}$$

$$W(y) = \frac{\partial^2 U(y)}{\partial \lambda^2} = \mathrm{Var}\left[Z(y)_{\lambda Z(y)}\right] = \sum_{\substack{1\le i\le d\\ y_i\neq b}} \frac{r_i(1-r_i)[\ln p_{i,y_i*}]^2}{\left[(1-r_i)p_{i,y_i*}^{\lambda} + r_i\right]^2} \tag{154}$$

the reverse Chernof inequality can be rewritten as

$$\ln \mathrm{Prob}\left\{Z(y) \ge V(y) - \sqrt{2W(y)}\right\} \ge U(y) - \lambda V(y) - \ln 2 - \lambda\sqrt{2W(y)} \tag{155}$$

It is time for the second inequality tier. For any $\delta < 1/3$

$$\mathrm{Prob}\Big\{|U(y) - \mathrm{E}[U]| \le \sqrt{\mathrm{Var}[U]/\delta}, \tag{156}$$

$$|V(y) - \mathrm{E}[V]| \le \sqrt{\mathrm{Var}[V]/\delta},\ W(y) \le \mathrm{E}[W]/\delta\Big\} \ge 1 - 3\delta \tag{157}$$

where

$$\mathrm{E}[U] = \sum_{i=1}^{d}\sum_{j=0}^{b} p_{i,j} \ln[1 - r_i + (j \neq b) r_i p_{i,j*}^{-\lambda}] \tag{158}$$

$$\mathrm{E}[V] = \sum_{i=1}^{d}\sum_{j=0}^{b-1} p_{i,j} \frac{-r_i \ln p_{i,j*}}{(1-r_i)p_{i,j*}^{\lambda} + r_i} \tag{159}$$

Hence

$$\ln \mathrm{Prob}\Big\{Z(y) \ge \mathrm{E}[V] - \sqrt{\mathrm{Var}[V]/\delta} - \sqrt{2\mathrm{E}[W]}/\delta\Big\} \ge \tag{160}$$

$$\ge \mathrm{E}[U] - \lambda \mathrm{E}[V] - \ln 2 - \sqrt{\mathrm{Var}[U]/\delta} - \lambda\sqrt{\mathrm{Var}[V]/\delta} - \lambda\sqrt{2\mathrm{E}[W]/\delta} \tag{161}$$

Now we have to pull all strings together. In order to connect with (146) we will require

$$\mathrm{E}[V] = (1+\epsilon)\ln n \tag{162}$$

$$\sqrt{\mathrm{Var}[U]} \le \epsilon \delta^{1/2} \lambda \ln n \tag{163}$$

$$\sqrt{\mathrm{Var}[V]} + \sqrt{2\mathrm{E}[W]} \le \epsilon \delta^{1/2} \ln n \tag{164}$$

for some small $\epsilon > 0$. Recalling (153), condition (162) is achieved by choosing λ to attain

$$\min_{\lambda \ge 0} [-(1+\epsilon)\lambda \ln n + \mathrm{E}[U]] \tag{165}$$

If (164) holds, then with probability at least $1 - 3\delta$

$$\ln S(y) \ge -(1+3\epsilon)\lambda \ln n + \mathrm{E}[U] - \ln 2 \tag{166}$$

Recalling (149) the success probability is at least

$$S \ge \frac{1-3\delta}{1 + 2e^{(1+3\epsilon)\lambda \ln n - \mathrm{E}[U]}/T} \tag{167}$$

■