CSE 521 Problem Set 2

Tyler Chen

Problem 1

Suppose we have a universe U of elements. For $A, B \subset U$, the Jaccard distance of A, B is defined as,

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

This definition is used practice to calculate a notion of similarity of documents, webpages, etc. For example, suppose U is the set of English words, and any set A represents a document considered as a bag of words. Note that for any two $A, B \subset U$, $0 \leq J(A,B) \leq 1$. If $J(A,B)$ is close to 1, then we say $A \approx B$.

(a) Let $h : U \to [0,1]$ where for each $i \in U$, $h(i)$ is chosen uniformly and independently at random. For a set $S \subset U$, let $h_S := \min_{i \in S} h(i)$. Show that,

$$\mathbb{P}[h_A = h_B] = J(A,B)$$

(b) Now, suppose we have sets $A_1, A_2, \ldots, A_n$, we can use the above idea to output the Jaccard similarity of all pairs of sets. In the input files `j1.in`, `j2.in`, `j3.in`, `j4.in` you are given the description of n sets. The first line of the the input contains n followed by $|U|$. The elements in each set are a subset of $\{1, \ldots, |U|\}$. In the next n lines, each line has the list of numbers in one of the sets. For all $1 \leq i, j \leq n$, in the $n(i-1)+j$ line of the output you should write the Jaccard similarity of the i-th and j-th set within 1 ± 0.1 multiplicative error, except for `j4.in` for which it is enough to get an write down the Jaccard similairity within 1 ± 0.5. The input file `j4.in` has only 10 percent of the grade.
Below you can see a sample input and output files. Upload your code together with all output files to Canvas. You will receive full grade of each test case as long as you get 90 percent of the numbers

```
j0.in          j0.out
3 6            0.21
1 6 4          0.49
3 2 6          0.2
1 2 4
```

Note that the correct Jaccard distances are 0.2, 0.5, 0.2 but it is enough to estimate the distance within 1 ± 0.1 multiplicative error, so you may output 0.21 instead of the correct distance of 0.2. Note that the naive algorithm would take $\mathcal{O}(n^2|U|)$ to calculate all pairwise similarities.

Solution

(a) Fix $A, B \subset U$ and let $i^* = \text{argmin}_{i \in A \cup B} h(i)$. Note that $h(i^*)$ has measure zero in $[0,1]$. The $h(i)$ are all independently chosen uniformly from $[0,1]$ so $\mathbb{P}[h(i) = h(i^*)] = \mathbb{1}[i = i^*]$ (note that we have assumed that U is countable).

This means $h_A = h_B$ if and only if i^* is in $A \cap B$.

Every points in $A \cup B$ has equal probability of being the argmin, so the probability that the argmin over the union is contained in the intersection is $J(A,B)$.

(b) We import the data using numpy.

```
# load file
z=0
f = np.fromfile('j'+str(z)+'.in',sep=' ',dtype='int')

# format data
n,size_U = f[:2]
A = np.reshape(f[2:],(n,-1))-1
del(f)

unique = np.unique(A)
size_unique = len(unique)
```

In the input data the sets have the same length so it is easy to store all the sets as a rectangular array. We note that there are some duplicate entries.

We first implement an exact algorithm using the `set` class in Python. The `set` class allows us to compute intersections very quickly.

```
A_set = []
for i in range(n):
    A_set.append(frozenset(A[i]))

jac_exact = []
for i in range(n):
    if i%10==0:
        print(i)
    for j in range(i+1,n):
        size_intersection = len(A_set[i]&A_set[j])
        size_union = len(A_set[i]|A_set[j])
        jac_exact.append( size_intersection/size_union )
```

Importing the data from `j4.in` and building the sets takes about 2 min, and computing the exact distances for all pairs of sets takes about 3 min.

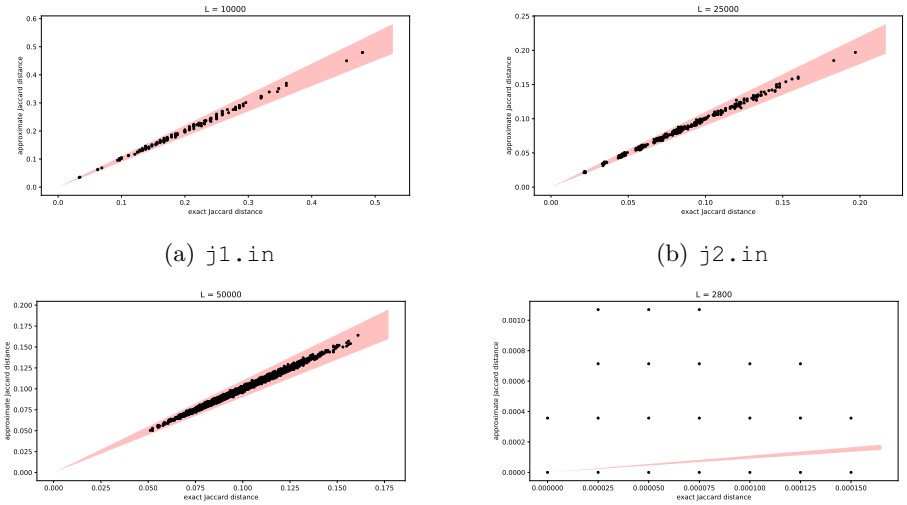
We now implement an approximate distance calculator based on the result in (a). In particular, we construct L independent hash functions $\{h^i\}_{i=1}^L$, and then hash each set in the data files. For any two sets A and B we compare $h^i(A)$ and $h^i(B)$, and compute what percent of the hash functions agree for a given set.

```
for l in range(L):
    if l%50==0:
        print(l)

    # generate hash function
    h[unique] = np.random.rand(size_unique)
    h_A[:,l] = np.min(h[A],axis=1)

jac_hash=[]
for i in range(n):
    for j in range(i+1,n):
        collisions = (h_A[i]==h_A[j])
        jac_hash.append(np.mean(collisions))
```

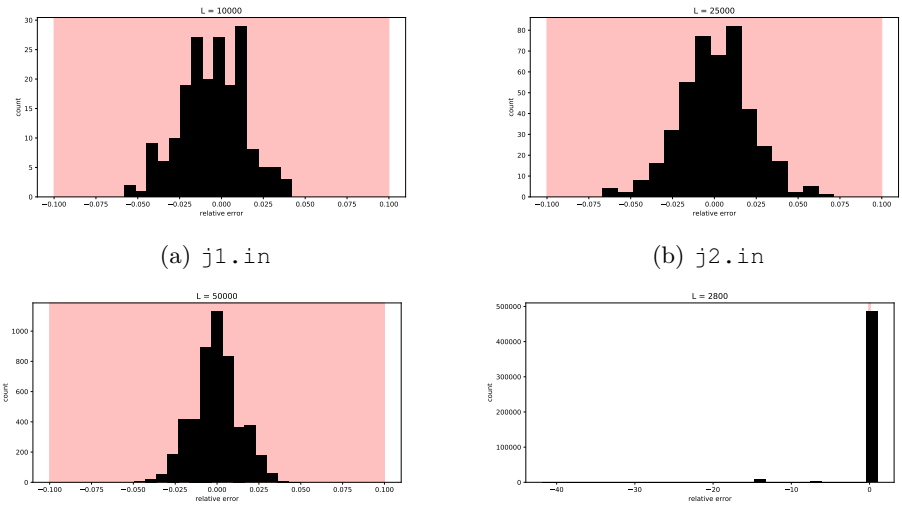
The problem with such an approach is that there are some set A and B such that $J(A,B)$ is very small but nonzero. Therefore, to even see a collision between some $h^i(A)$ and $h^i(B)$ for some i, we need L to be very large (i.e. $L > 1/J(A,B)$). This is not tractable in the case of `j4.in` since computing a hash function requires first generating $|\cap_i A_i|$ random numbers (about 20 million), and then computing the min over each set. This takes more than half a second, so to compute thousands of hash functions is not reasonable.



(a) `j1.in` (b) `j2.in`

(c) `j3.in` (d) `j4.in`

Figure 1: actual vs approximate Jaccard distances with 0.1 multiplicative and additive errors highlighted.



(a) `j1.in` (b) `j2.in`

(c) `j3.in` (d) `j4.in`

Figure 2: relative error counts with 0.1 multiplicative error highlighted.

Figure 1 shows the actual and approximate Jaccard distances over all pairs of sets. The red highlighted regions are 0.1 multiplicative and additive errors.

Figure 2 shows a histogram of the relative errors of the Jacard distances. The red highlighted regions are 0.1 multiplicative errors.

We see that for the first three data files we are within 0.1 multiplicative error.

For the `j4.in` we are definitely not in this range. Note that the number of hash functions L must increase with the universe size in order to be able to compute the decreasing Jaccard distances. This would not have necessarily been the case if the Jaccard distances between the sets did not decrease. Since all of the Jaccard distances are small we are trivially within a small additive error of the actual distances.

Problem 2

In this problem we design an LSH for points in $\mathbb{R}^d$, with the ℓ_1 distance, i.e.,

$$d(p,q) = \sum_i |p_i - q_i|$$

(a) Let a, b be arbitrary real numbers. Fix $w > 0$ and let $s \in [0, w)$ be chosen uniformly at random. Show that,

$$\mathbb{P}\left[\left\lfloor \frac{a-s}{w} \right\rfloor = \left\lfloor \frac{b-s}{w} \right\rfloor\right] = \max\left\{0, 1 - \frac{|a-b|}{w}\right\}$$

Recall that for any real number c, $\lfloor c \rfloor$ is the largest integer which is at most c.
Hint: Start with the case where $a = 0$.

(b) Define a class of has functions as follows: Fix w larger than diameter of the space. Each hash function is defined via a choice of d independently selected random real numbers $s_1, , s_2, \ldots, s_d$, each uniform in $[0, w)$. The hash function associated with this random set of choices is,

$$h(x_1, x_2, \ldots, x_d) = \left(\left\lfloor \frac{x_1 - s_1}{w} \right\rfloor, \left\lfloor \frac{x_2 - s_2}{w} \right\rfloor, \ldots, \left\lfloor \frac{x_d - s_d}{w} \right\rfloor\right)$$

Let $a_i = |p_i - q_i|$. What is the probability that $h(p) = h(q)$ in terms of the a_i values? For what values of p_1 and p_2 is this family of functions $(r, c \cdot r, p_1, p_2)$-sensitive? Do your calculations assuming that $1 - x$ is well approximated by e^{-x}.

Solution

(a) Pick A and B arbitrarily and fix $w > 0$. Let S be a uniform random variable on $[0, w)$. Now define,

$$a = A/w, \qquad b = B/w, \qquad s = S/w$$

Note that $\hat{s}$ is a uniform random variable on $[0, 1)$.

Therefore, it is sufficient to show,

$$\mathbb{P}[\lfloor a - s \rfloor = \lfloor b - s \rfloor] = \max\{0, 1 - |a-b|\}, \qquad s \sim \mathcal{U}([0,1))$$

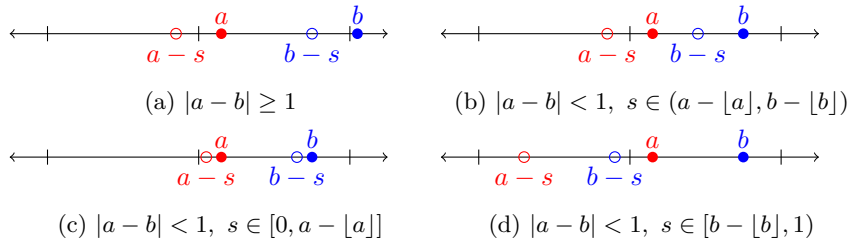
WLOG assume $a \le b$. Suppose $|a - b| \ge 1$. Then $\lfloor a - s \rfloor \neq \lfloor b - s \rfloor$ for any s. This is demonstrated in Figure 3a.

Now, suppose $|a - b| < 1$. Then $\lfloor a - s \rfloor \neq \lfloor b - s \rfloor$ if and only if $s \in (a - \lfloor a \rfloor, b - \lfloor b \rfloor)$. The three cases are shown in Figures 3b,3c,3d.

Now note that since $|a - b| < 1$, $\lfloor a \rfloor = \lfloor b \rfloor$. Therefore the interval $(a - \lfloor a \rfloor, b - \lfloor b \rfloor)$ has length exactly $|a - b|$. We choose s in this interval with probability $|a - b|/(1 - 0) = |a - b|$.

This proves,

$$\mathbb{P}\left[\left\lfloor \frac{A-S}{w} \right\rfloor = \left\lfloor \frac{B-S}{w} \right\rfloor\right] = \max\left\{0, 1 - \frac{|A-B|}{w}\right\}$$



(a) $|a-b| \geq 1$ (b) $|a-b| < 1,\ s \in (a - \lfloor a \rfloor, b - \lfloor b \rfloor)$

(c) $|a-b| < 1,\ s \in [0, a - \lfloor a \rfloor]$ (d) $|a-b| < 1,\ s \in [b - \lfloor b \rfloor, 1)$

Figure 3

(b) The event $h(p_i) = h(q_i)$ is independent of the event $h(p_j) = h(q_j)$ for $i \neq j$ since the s_i are independent. Moreover, since w is larger than the diameter of the space, $|p_i - q_i| < w$ so that $1 - a_i/w > 0$. Therefore,

$$\mathbb{P}[h(p) = h(q)] = \mathbb{P}\left[\forall i : h(p_i) = h(q_i)\right] = \prod_{i=1}^{d} \mathbb{P}\left[h(p_i) = h(q_i)\right] = \prod_{i=1}^{d} (1 - a_i/w)$$

We now make the assumption that $1 - a_i/w \approx e^{-a_i/w}$ so that,

$$\mathbb{P}[h(p) = h(q)] = \prod_{i=1}^{d} (1 - a_i/w) \approx \exp\left(-\sum_{i=1}^{d} \frac{a_i}{w}\right) \approx \exp\left(-\frac{d(p,q)}{w}\right)$$

where we have used the definition of a_i to write $d(p,q) = \sum_i a_i$.

If $d(p,q) \leq r$ then,

$$\mathbb{P}[h(p) = h(q)] \geq e^{-r/w}$$

Similarly, if $d(p,q) \geq c \cdot r$ then,

$$\mathbb{P}[h(p) = h(q)] \leq e^{-cr/w}$$

Therefore this family of has functions is $(r, c \cdot r, p_1, p_2)$-sensitive for,

$$p_1 = e^{-r/w}, \qquad p_2 = e^{-cr/w}$$

Note that $e^{-x} = 1 - x + x^2/2! + \mathcal{O}(x^3)$ so $e^{-x} \geq 1 - x$. Therefore our value of p_1 is actually lower than it should be.

Problem 3

Let $u, v \in \mathbb{R}^d$ and $g \in \mathbb{R}^d$ be a random Gaussian vector, i.e., for each $1 \leq i \leq d$, $g_i \sim \mathcal{N}(0,1)$.

(a) What is the expected value of $\langle g, u\rangle$?
(b) What is the expected value of $\langle g, u\rangle \cdot \langle g, v\rangle$?
(c) What is the expected value of $|\langle g, u\rangle|$? You can use the p.d.f.of a $\mathcal{N}(0,1)$ is $\frac{1}{\sqrt{2\pi}}e^{-x^2/2}$.
(d) Consider the following hash function: $h_g(u) = \text{sgn}(\langle g, u\rangle)$, where sgn is the sign function, i.e., $\text{sgn}(a) = 1$ if $a \geq 0$ and $\text{sgn}(a) = -1$ otherwise. Show that for a Gaussian random vector g and any two vectors u, v, $\mathbb{P}[h_g(u) = h_g(v)] = 1 - \frac{\theta(u,v)}{\pi}$ where $\theta(p,q)$ is the angle between the vectors p and q.
(e) Let $P \subseteq \mathbb{R}^d$ and consider the following discrete distance function: $\text{dist}(p,q) = \frac{\theta(p,q)}{\pi}$. For what values of p_1 and p_2 is this family of functions $(r, c\cdot r, p_1, p_2)$-sensitive? You can do your calculations assuming that $1 - x$ is well approximated by e^{-x}.

Solution

(a) We take $\langle\cdot,\cdot\rangle$ to be the Euclidean inner product. Then,

$$\mathbb{E}[\langle g, u\rangle] = \mathbb{E}\left[\sum_{i=1}^{d} u_i g_i\right] = \sum_{i=1}^{d} u_i \mathbb{E}[g_i] = 0$$

(b) Note that the entries of g are independent so that $\mathbb{E}[g_i g_j] = \mathbb{E}[g_i]\mathbb{E}[g_j] = 0$ if $i \neq j$. Note further that $\mathbb{E}[g_i^2] = \mathbb{V}[g_i] + \mathbb{E}[g_i]^2 = 1$. Therefore,

$$\begin{aligned}
\mathbb{E}[\langle g, u\rangle \cdot \langle g, v\rangle] &= \mathbb{E}\left[\left(\sum_{i=1}^{d} u_i g_i\right)\left(\sum_{j=1}^{d} v_j g_j\right)\right] \\
&= \mathbb{E}\left[\sum_{i=1}^{d}\sum_{\substack{j=1\\ j\neq i}}^{d} u_i v_i g_i g_j + \sum_{i=1}^{d} u_i v_i g_i^2\right] \\
&= \sum_{i=1}^{d}\sum_{\substack{j=1\\ j\neq i}}^{d} u_i v_i \mathbb{E}[g_i g_j] + \sum_{i=1}^{d} u_i v_i \mathbb{E}[g_i^2] \\
&= 0 + \sum_{i=1}^{d} u_i v_i \\
&= \langle u, v\rangle
\end{aligned}$$

(c) From (a) and (b) we know that $\langle g, u\rangle$ has mean 0 and variance $\mathbb{E}[\langle g, u\rangle^2] - \mathbb{E}[\langle g, u\rangle]^2 = \langle u, u\rangle = \|u\|^2$. Moreover, by construction $\langle g, u\rangle$ is the sum of normal random variables and therefore is a normal random variable. That is, $\langle g, u\rangle \sim \mathcal{N}(0, \|u\|^2)$.

Let $X \sim \mathcal{N}(0, \sigma^2)$. Then X has density function,

$$\phi(z) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(\frac{-z^2}{2\sigma^2}\right)$$

We find the density function ψ of $|X|$. First, observe,

$$\mathbb{P}[|X| \leq x] = \mathbb{P}[-x \leq X \leq x] = \int_{-x}^{x} \phi(z)\mathrm{d}z$$

Therefore, the density function is,

$$\psi(x) = \frac{\mathrm{d}}{\mathrm{d}x}\mathbb{P}[|X| \leq x] = \frac{\mathrm{d}}{\mathrm{d}x}\int_{-x}^{x} \phi(z)\mathrm{d}z = \phi(x) + \phi(-x) = 2\phi(x)$$

Finally using standard facts about the Gaussian integral,

$$\mathbb{E}[|X|] = \int_0^\infty z\psi(z)\mathrm{d}z = \frac{2}{\sqrt{2\pi\sigma^2}}\int_0^\infty \exp\left(\frac{-z^2}{2\sigma^2}\right) = \sqrt{\frac{2\sigma^2}{\pi}}$$

Therefore,

$$\mathbb{E}[|\langle g, u\rangle|] = \sqrt{\frac{2\|u\|^2}{\pi}} = \sqrt{\frac{2}{\pi}}\|u\|$$

(d) Fix u, v and let g be a Gaussian random vector.

Let $\mathcal{H}$ be the unique hyperplane containing u and v. Let $\mathcal{H}_u = \{x \in \mathcal{H} : \langle x, u\rangle \geq 0\}$ and let $\mathcal{H}_v = \{x \in \mathcal{H} : \langle x, v\rangle \geq 0\}$.

Then for any $x \in \mathcal{H}$, $h_u(x) = h_v(x)$ if and only if $x \in \mathcal{H}_u \cap \mathcal{H}_v$ or $x \notin \mathcal{H}_u \cup \mathcal{H}_v$. This is illustrated in Figure 4.

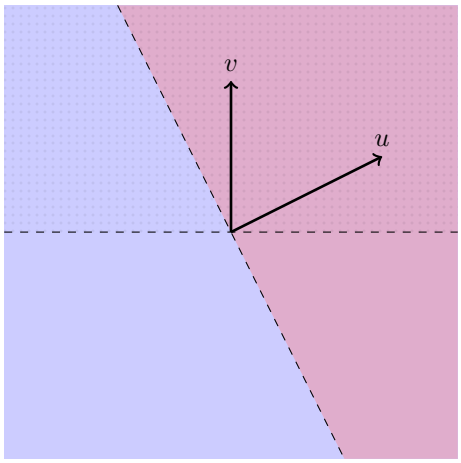


Figure 4: Hyperplane defined by u and v. $\mathcal{H}_u$ is the dotted region, and $\mathcal{H}_v$ is the red region.

Let $\hat{g}$ be the orthogonal projection of g onto $\mathcal{H}$. Note that $\langle g, x\rangle = \langle \hat{g}, x\rangle$ for all $x \in \mathcal{H}$ since x is orthogonal to everything outside of the hyperplane.

By the rotational invariance of Gaussian random vectors, the angle $\hat{g}$ makes with any fixed vector in $\mathcal{H}$ is uniformly distributed on $[0, 2\pi)$.

From Figure 4 it is clear that the chance of $\hat{g}$ landing in $\mathcal{H}_u \cup \mathcal{H}_v$ is $(\pi - \theta)/2\pi$ and the chance of $\hat{g}$ landing in $(\mathcal{H}_u \cap \mathcal{H}_v)^c$ is also $(\pi - \theta)/2$.

Therefore, the chance of $\hat{g}$ landing in $\mathcal{H}_u \cap \mathcal{H}_v$ or $(\mathcal{H}_u \cup \mathcal{H}_v)^c$ is $1 - \theta/\pi$. That is,

$$\mathbb{P}[h_g(u) = h_g(v)] = 1 - \frac{\theta(u,v)}{\pi}$$

(e) If $\theta(u,v)/\pi = \text{dist}(u,v) \leq r$ then,

$$\mathbb{P}[h_g(u) = h_g(v)] = 1 - \text{dist}(u,v) \geq 1 - r$$

Similarly, if $\theta(u,v)/\pi = \text{dist}(u,v) \geq c \cdot r$ then,

$$\mathbb{P}[h_g(u) = h_g(v)] = 1 - \text{dist}(u,v) \leq 1 - cr$$

Therefore this family of has functions is $(r, c \cdot r, p_1, p_2)$-sensitive for,

$$p_1 = 1 - r, \qquad p_2 = 1 - cr$$

Problem 4

Describe an example (i.e., an appropriate set of points in $\mathbb{R}^n$) that shows that the Johnson-Lindenstrauss dimension reduction method, the linear transformation obtained by projecting on Gaussian vectors scaled properly, does not preserve ℓ_1 distances within even factor 2.

Solution

Define $G \in \mathbb{R}^{k\times n}$ as,

$$G = \frac{1}{k}\sqrt{\frac{\pi}{2}}\begin{bmatrix} - g^1 - \\ - g^2 - \\ \vdots \\ - g^k - \end{bmatrix}$$

Now note that for any vector x,

$$Gx = \frac{1}{k}\sqrt{\frac{\pi}{2}}\begin{bmatrix} \langle g^1, x\rangle \\ \langle g^2, x\rangle \\ \vdots \\ \langle g^k, x\rangle \end{bmatrix}$$

Therefore, since $\mathbb{E}[|\langle g^i, x\rangle|] = \sqrt{2/\pi}\,\|x\|_2$,

$$\mathbb{E}\left[\|Gx\|_1\right] = \frac{1}{k}\sqrt{\frac{\pi}{2}}\sum_{i=1}^{k}\mathbb{E}\left[\left|\langle g^i, x\rangle\right|\right] = \frac{1}{k}\sqrt{\frac{\pi}{i}}\sum_{i=1}^{k}\sqrt{\frac{2}{\pi}}\,\|x\|_2 = \|x\|_2$$

It is then clear that G (or any scalar multiple of G) will not preserve the relative ℓ_1 norms of vectors they act on. This means that they will not preserve distances, since the norm of a vector is the distance to zero.

For example, consider $u \in \mathbb{R}^n$, and $v \in \mathbb{R}^n$ defined as,

$$u = \begin{bmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix}, \qquad\qquad v = \frac{1}{n}\begin{bmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{bmatrix}$$

Then $\|u\|_1 = \|v\|_1 = 1$ but $\|u\|_2 = 1$ while $\|v\|_2 = \sqrt{n}$ so whatever constant we put in front of G to preserve the norm of u will not preserve the norm of v.

Therefore, if we preserve (in expectation) the ℓ_1 distance from u to 0, we cannot preserve (in expectation) the ℓ_1 distance from v to 0.