Problem 1: The Exacting LSH

(i) Assume that there is some exact-LSHable similarity function \( \text{sim} \) and points \( x, y, z \) for which \( d(x, y) > d(x, z) + d(z, y) \). Then:

\[
\begin{align*}
d(x, y) &> d(x, z) + d(z, y) \\
1 - \text{sim}(x, y) &> 1 - \text{sim}(x, z) + 1 - \text{sim}(z, y) \\
1 - \Pr(h(x) = h(y)) &> 2 - \Pr(h(x) = h(z)) - \Pr(h(z) = h(y)) \\
\Pr(h(x) \neq h(y)) &> 2 - (\Pr(h(x) = h(z)) + \Pr(h(z) = h(y))) \\
\Pr(h(x) \neq h(y)) &> 2 - \Pr(h(x) = h(z) = h(y))
\end{align*}
\]

since by the union bound \( \Pr(h(x) = h(z) = h(y)) \leq \Pr(h(x) = h(z)) + \Pr(h(z) = h(y)) \). But then \( \Pr(h(x) \neq h(y)) > 1 \), which is nonsensical, so no such points exist.

(ii)

\[
\begin{align*}
d(\{1\}, \{2\}) &= 1 - \text{sim}_{\text{Dice}}(\{1\}, \{2\}) = 1 - \frac{|\{1\}|}{\frac{1}{2}(|\{1\}| + |\{2\}|)} = 1 \\
d(\{1\}, \{1, 2\}) &= 1 - \text{sim}_{\text{Dice}}(\{1\}, \{1, 2\}) = 1 - \frac{|\{1\}|}{\frac{1}{2}(|\{1\}| + |\{1, 2\}|)} = \frac{1}{3} \\
d(\{2\}, \{1, 2\}) &= 1 - \text{sim}_{\text{Dice}}(\{2\}, \{1, 2\}) = \frac{1}{3}
\end{align*}
\]

so that \( d(\{1\}, \{2\}) > d(\{1\}, \{1, 2\}) + d(\{1, 2\}, \{2\}) \).

(iii)

\[
\begin{align*}
d(\{1\}, \{2\}) &= 1 - \text{sim}_{\text{over}}(\{1\}, \{2\}) = 1 - \frac{|\{1\}|}{\min(|\{1\}|, |\{2\}|)} = 1 \\
d(\{1\}, \{1, 2\}) &= 1 - \text{sim}_{\text{over}}(\{1\}, \{1, 2\}) = 1 - \frac{|\{1\}|}{\min(|\{1\}|, |\{1, 2\}|)} = 0 \\
d(\{2\}, \{1, 2\}) &= 1 - \text{sim}_{\text{over}}(\{2\}, \{1, 2\}) = 1 - \frac{|\{2\}|}{\min(|\{2\}|, |\{1, 2\}|)} = 0
\end{align*}
\]

so that \( d(\{1\}, \{2\}) > d(\{1\}, \{1, 2\}) + d(\{1, 2\}, \{2\}) \).

Problem 2: Finding Your Neighbors

(i) We have

\[
\mathbb{E} \left[ \sum_{j=1}^{L} |T \cap W_j| \right] = \sum_{j=1}^{L} \mathbb{E}[T \cap W_j] = L \mathbb{E}[T \cap W_i]
\]

1
since the $g_j$ are iid. Thus, using Markov’s inequality, we only need to show $\mathbb{E} |T \cap W_1| \leq 1$. Now,

$$\mathbb{E} |T \cap W_1| = \sum_{x \in A} \mathbb{E} (I(x \in T \cap W_1))$$

$$= \sum_{x \in A} \Pr (d(x, z) > c\lambda, g_1(x) = g_1(z))$$

$$= \sum_{x \in T} \Pr (g_1(x) = g_1(z))$$

$$\leq \sum_{x \in T} (p_2)^k$$

since for each of the $k$ independent hash functions in $g_1$, the collision probability for points more than $c\lambda$ distance from $z$ is at most $p_2$. But $k = \lceil \log_1/p_2(n) \rceil$, so $p^k_2 = \left(\frac{1}{p_2}\right)^{-k} \leq \left(\frac{1}{p_2}\right)^{- \log_1/p_2(n)} = \frac{1}{n}$.

Thus $\mathbb{E} |T \cap W_1| \leq \frac{|T|}{n} \leq 1$.

(ii) For $d(x^*, z) \leq \lambda$, we have

$$\Pr [g_j(x^*) \neq g_j(z) \forall 1 \leq j \leq L] = \Pr [g_1(x^*) \neq g_1(z)]^L$$

$$= (1 - \Pr [g_1(x^*) = g_1(z)])^L$$

$$= \left(1 - \Pr_{h \leftarrow H} [h(x^*) = h(z)]^k\right)^L$$

$$\leq \left(1 - p_1^k\right)^L$$

$$< \left(e^{-p_1^k}\right)^L = e^{-p_1^kL}.$$

Now, to find the value of that exponent, consider

$$\log(p_1^kL) = k \log p_1 + \log L = \frac{\log n}{\log 1/p_2} \log p_1 + \frac{\log 1/p_1}{\log 1/p_2} \log n = 0.$$

Thus $p_1^kL = 1$, and

$$\Pr [g_j(x^*) \neq g_j(z) \forall 1 \leq j \leq L] < \frac{1}{e}$$

as desired.

(iii) The reported point is a $(c, \lambda)$-ANN iff one was selected among our sample of size $3L$ from $\bigcup_j W_j$.

Part (i) says that with probability at least $\frac{2}{3}$, $\bigcup_j W_j$ contains fewer than $3L$ “bad points”; if so, we will report a $(c, \lambda)$-ANN if there are any in $\bigcup_j W_j$.

Let the number of $(c, \lambda)$-ANNs in $A$ be $m$; we’re guaranteed that $m \geq 1$. Part (ii) shows that each such point is in $\bigcup_j W_j$ with probability at least $1 - \frac{1}{e}$. Thus the probability that any of them fall in $\bigcup_j W_j$ is at least $1 - e^{-m}$. For $m = 1$, this is about .63; for $m = 3$, this is already over .95.

Thus, we make an error with probability at most $\frac{1}{3} + \frac{1}{e}$; so with probability at least about 0.3 we will return a $(c, \lambda)$-ANN. Though this number is pretty low, note that the first part of the bound is quite loose and the second also increases quickly with the number of $(c, \lambda)$-ANNs.
Mean search time is about 0.013 seconds using LSH with $L = 10$, $k = 24$ on my desktop, and about 0.14 seconds using linear search.


Error goes down more or less monotonically while increasing $L$; query time is not strongly affected. Increasing $k$ substantially increases error and also substantially decreases query time.

![Figure 1: Distance inflation and query time.](image1.png)

(a) varying $L$ with $k = 24$  
(b) varying $k$ with $L = 10$

The nearest neighbors for the two methods are visually quite similar, though you can see that the third NN with LSH is already the eighth true NN.

![Figure 2: Nearest neighbor results: first row is linear search solutions, second row is LSH with $L = 10$ and $k = 24$, with distance increasing to the right.](image2.png)