一、traditional method: (Gram)

For J: USERS K: RE M: Words

$x_j \in C^{K \times M}$: Codebook for j_{th} user

For Column m, the codewords is $c_{j,m}$.

The signal is: $x_j = diag(w_j) c_{j,m} \in C^K$

The superposed codewords: $x(m_1, m_2 \cdots m_J) = \sum_{j=1}^{J} diag(w_j) c_{jm_j}$

So. there's M^J possibilities superposed words

1、Each of the superposed have J users of K dimensions.

So the complexity is: $O(M^J \cdot J \cdot K)$

2、$||x_p - x_q||^2 = ||x_p||^2 + ||x_q||^2 - 2R(x_p^H x_q)$.

Gram matrix $G = XX^H \in C^{N \times N} \longrightarrow G_{pq} = x_p^H x_q$

↑ superposed words

↑ 共轭转置

the p_{th} element of vector $n_p = \|x_p\|_2^2 = G_{pp}$

$$D^2 = n\mathbf{1}^T + \mathbf{1}n^T - 2R(G)$$

$O(M^{2J}K)$: $G_{pq} = \sum_{k=1}^{K} x_{p,k}\,\overline{x_{q,k}}$

K times product, and M^J time in total.

So the time complexity

overall is : $O(M^J \cdot J \cdot K + M^{2J}K)$ (KD-tree)

$\|$

$O(M^{2J}K) \rightarrow (M^J \cdot J \log M \cdot K)$

二、 assume that $s(m) = \sum_{j=1}^{J} x_{j,m_j}$ (superposed), $m = (m_1, \ldots m_J)$.

$$MED_{sup} = \min \|s(m) - s(n)\|_2$$

For $m_u \neq n_u$ $\quad m_j = n_j$

there's : $S(m) = \sum_{j \neq u} x_{j,m_j} + x_{u,m_u}$.

$$S(n) = \sum_{j \neq u} x_{j,n_j} + x_{u,n_u}$$

So, $S(m) - S(n)$ can always be:

$$x_{u,m_u} - x_{u,n_u}$$

No matter what is the j',

the distance can always be: $x_{u,m_u} - x_{u,n_u}$

Complexity : for every user, pick up

M^2 codewords, and for every codeword, needs one calculation for distance.

So, $O(j \cdot K \cdot M^2)$

Can be applied to: Sparse Codebook

$MED \leq d_{min}$

3. For 2 users.

Hamming Distance → $d_H(m,n) = \#\{j : m_j \neq n_j\}$

All pairs: $P = \{(m,n) \in \Omega^2 : m \neq n\}$

$P_k = \{(m,n) \in P : d_H(m,n) = k\}$

We have $P_1 \subseteq P$, $P_2 \subseteq P$, $(P_1 \cup P_2) \subseteq P$.

Assume that $d_{H_2} = d_{min}(P_1, P_2)$

there is $d_{min} = \min_{P} d \le \min_{P_1 \cup P_2} d \le \min_{P_1} d$

So, $MED \le d_{H_2 \cup H_1} \le d_{H_1}$

that is, more users in error can make the result convergence to the MED. But consumes more complexity.

Calculation of Complexity:

For a user, it's $O(M^2)$

So 2 users, there's O^2 possibilities

so, it comes to $O(M^4)$

And there is $\binom{J}{2}$ user pairs.

So complexity overall is $\bar{J}^2$

$O(M^4\bar{J}^2K)$, For $O_{H_1 \cup H_2}(M^4\bar{J}^2K + M^2\bar{J}K)$

What about using a tree structure?

KD-tree in Data Structure. is likely to reduce the complexity

4. KD-tree + Hamming-Distance ≥ 2.

For Hamming-Distance ≥ 2

$$S(m) - S(n) = (X_{i,m_i} - X_{i,n_i}) + (X_{j,m_j} - X_{j,n_j})$$

$$\Delta_i = X_{i,a} - X_{i,b}, \quad \Delta_j = X_{j,p} - X_{j,q}$$

So MED: $d_{H_2} = \underbrace{\min_{i<j}}_{\text{}} \min_{\Delta_i \in D_i, \Delta_j \in D_j} \|\Delta_i + \Delta_j\|$

Different user pairs, avoid repeat

For $||\Delta j - (-\Delta i)||$ —> crucial step

That is to say, for certain Δi, which wants $||\Delta i + \Delta j||$ to be smallest. we need to find the closest Δj to the point $-\Delta i$

Then, Due to the KD-tree can be only inplemen on Real numbers,

For $\Delta j = (a_1 + b_1 j, \cdots, a_K + b_K j)$

we change it into a $2K$-dimension

$$z_j = (a_1, b_1, \cdots, a_k, b_k) \in R^{2k}$$

The distance can be :

$$\|\Delta - \Delta'\|_2^2 = \sum_{k=1}^{K} |(a_k + ib_k) - (a'_k + ib_k')|^2$$

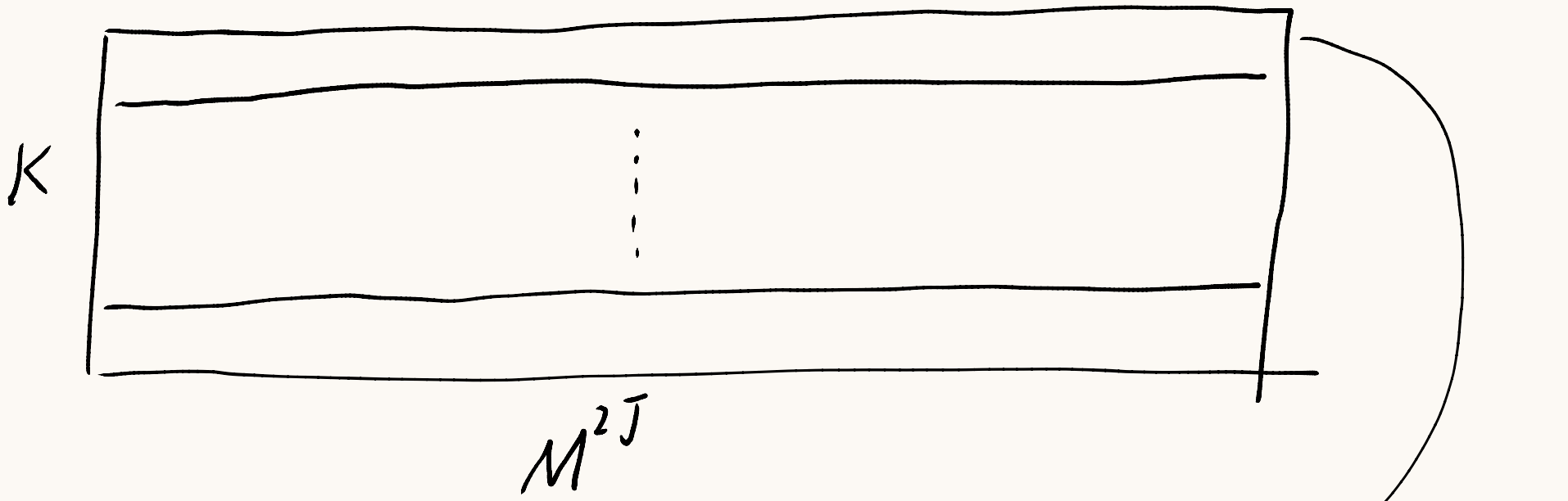
$$\Leftrightarrow \sum_{k=1}^{K} [(a_k - a'_k)^2 + (b_k - b_k')^2]$$

↓

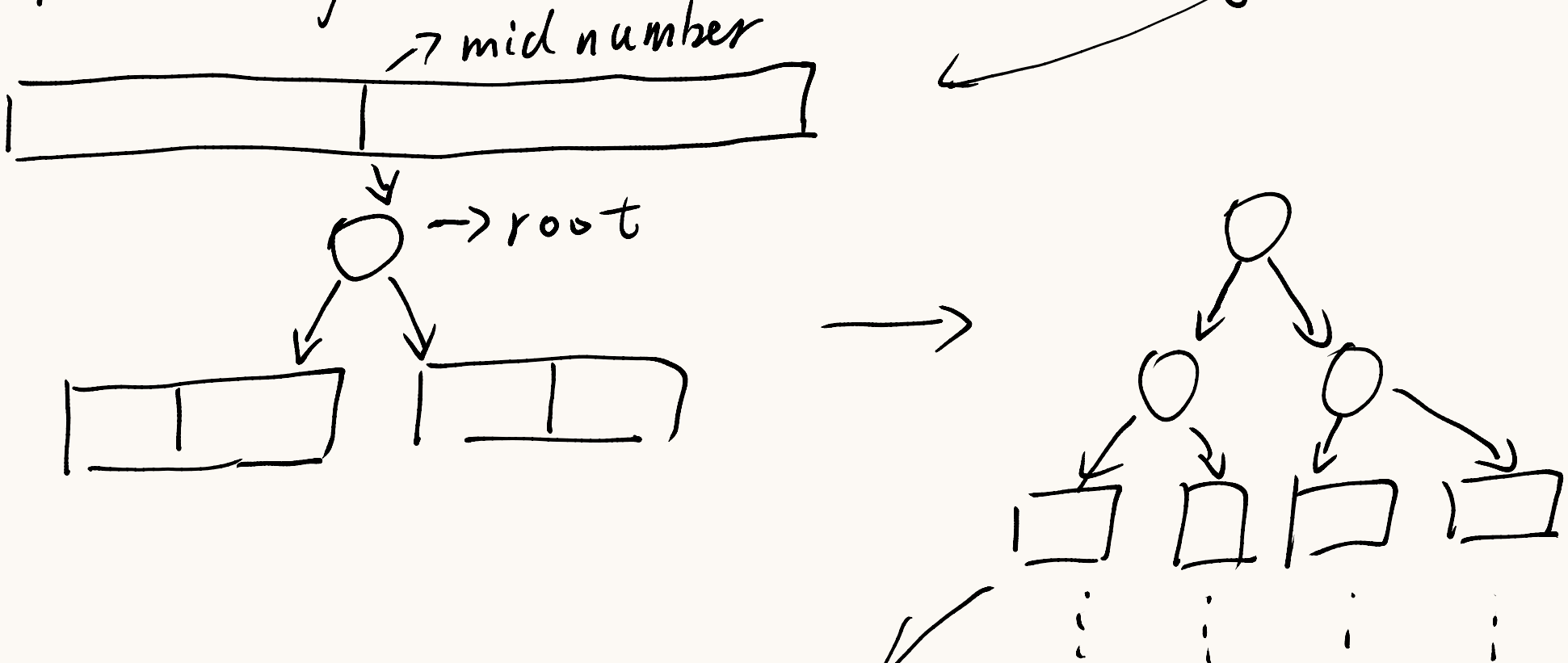
This is result is the same as Distance with complex number.

Then, when it comes to the trees.

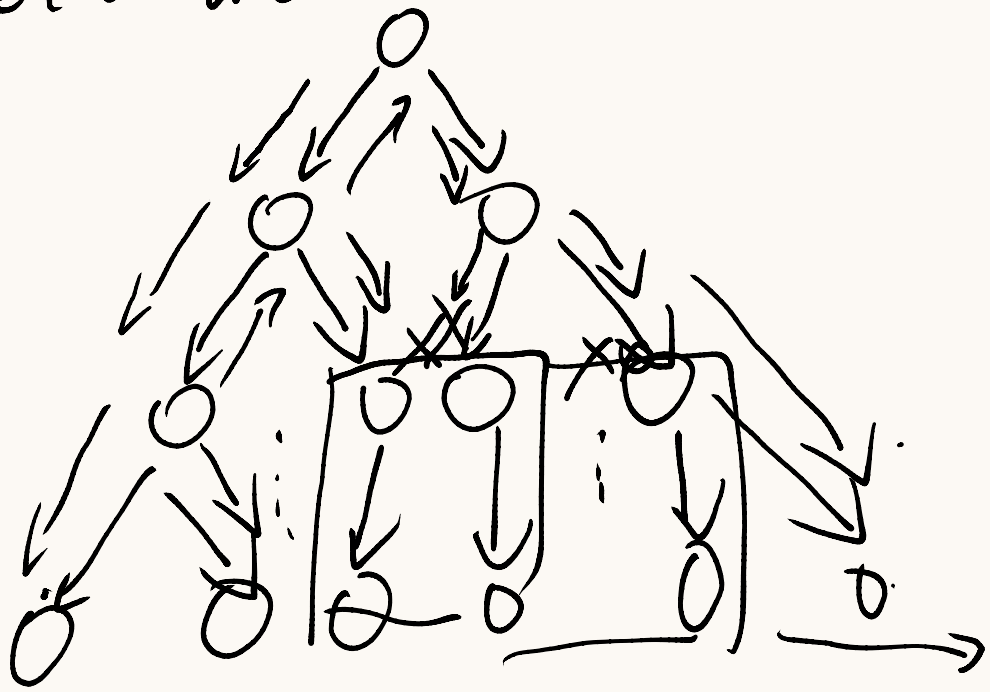
For an example



For a singer dimension. (For small to big)



Lastly, we get a tree.



save the time

Now we have a D_i (difference) for user i
and a tree a user j

for $\Delta_i \in D_i$, extend it to a real vector z_i,
and get it as $-z_i$, which correspond to $-\Delta_i$
in the complex num.

put it as a query point $q = -z_i$

and perform a nearest neighbor search in a KD-tree returning with a nearest $z_j^{(q)}$

Since $\|z_j^{(q)} - q\| = \|z_j^{(q)} - (-z_i)\|$

$\|\Delta_i + \Delta_j\|$

which is d_{1+2}

best, worst M.

For nearest neigbor query, its average search complex $O(\log M)$.

So overall complexity can be about

Build $\leftarrow O(JM^2K\log M) + O(J^2M^2K\log M)$

can be $(JMK\log M)$ $\leftarrow$ $O(J^2M^2K\log M)$

So overall, $O_{H_1 UH_2}(J^2M^2K\log M + M^2JK)$

if more needs more accuracy, the

$O_{H_1 UH_2 UH_3}(J^3M^3K\log M + \cdots)$ (concluded)

care about the space complexity! more trees needed!

5. what about sampling? with KD-tree?

For picking S superposed words, $\rightarrow O(SJK)$

and calculating the distance $\rightarrow O(S^2K)$

That is $O(sJK + s^2K)$
or $O(sJK + SK \log S)$ (KD-tree)

Not for big-scale too complex

~~Uniform Grid?~~ ~~Closest pair?~~

Early abandon?

Cauchy-Schwarz?

For Hamming Distance, Sample.,
they can gave lots of caculation. But,
this is sacrifice lots of accuracy (maybe).

An availabe access to this method is:

Hamming 1+2 + KD-tree Sampling.

take the smallest distance as the result.

For traditional:

1. KD-tree

$O(M^{2J}K) \rightarrow (MJK \log M^J)$

2. set a d_{best} at every caculation.

if $\sum_{k=1}^{t} (x_k - y_k)^2, t = 1 \cdots k.$

if one of them is bigger the d_{best}.

just abandon it.

3. lower bound pruning (Cauchy-Schwarz)

$$\|x-y\| \geq \|x\| - \|y\|$$

try $\|x\| - \|y\|$? d_{best}.

if $\|x\| - \|y\| \geq d_{best}$.

just abandon it.

4. devided Gram