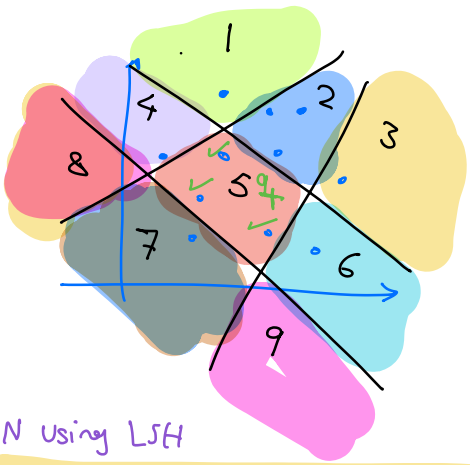
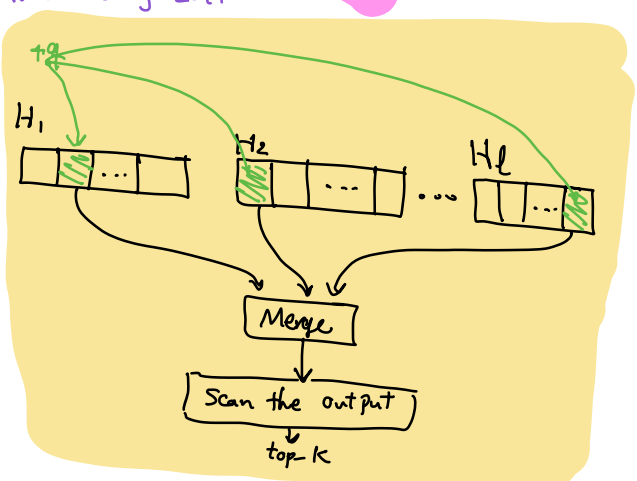
LSH for Euclidean distance (l-2 norm)



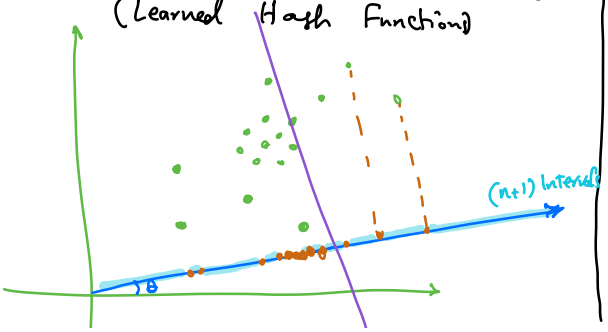
ANN using LSH



LSH Limitations

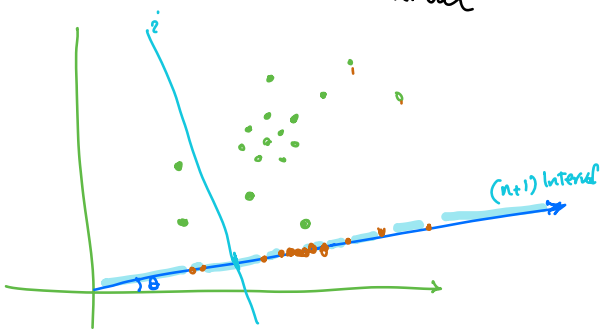
1. Data Uninformed => Dense cells can have a lot of points
2. Memory => Hash functions require a large storage.

Resolution 1 : Data-Informed Hash Functions (Learned Hash Functions)



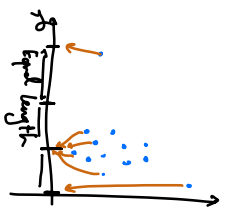
To draw a data Informed Hash Function

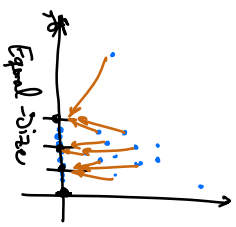
1. draw a random angle θ
2. divid the projections of Points into (n+1) intervals
3. draw a random number [0, n]
4. draw a Line orthogonal to θ of the Selected Interval



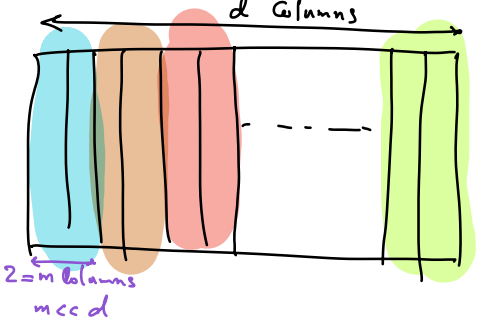
i ← Rand(0, n)

Quantization





Product Quantization



Step 1: Create Column blocks by grouping every $m \ll d$ columns
$\Rightarrow d/m$ Column blocks

Step 2: quantize the data for every c-block into k values

e.g.: $k = 32 \Rightarrow \log_2 32 = 5$ bits

e.g., $t = \langle 32.6, -8, 5, 4.3, \dots, 8.1, 9.32 \rangle$

$PQ\text{-Code}(t) = [\emptyset \in [32], 3, \dots, 30]$

$|PQ\text{-Code}(t)| = d/m \log_2 k$

* Cluster the data on the selected Column block into k clusters using k-means clustering
* the Centroid of each cluster is the representative of it

e.g., Column (0,1) $\Rightarrow$ $0 \rightsquigarrow \mu_0(30,-7)$, $1 \rightsquigarrow \mu_1(\cdot,\cdot)$, 2, 3 ...

$PQ(t) = i(t) = [i^0(t), i^1(t), \dots i^{(m)}(t)]$

Position Approximation

e.g., $t \rightsquigarrow i(t) = [\emptyset, 3, 8, 30]$

$\tilde{t} = i^{-1}(t) = [30, -7, 5, 4, \dots]$

$e^2(t) = \sum_{j=1}^{d} (t_j - \tilde{t}_j)^2 \le \sum_{j=1}^{d/m} \delta(t, \mu^j(t))^2$

- to Construct a LSH-Style PQ
  - Shuffle the Columns l times & build a PQ-code for each Shuffle