MTBF (Mean Time Between Failure)

System with N components $C_1, \dots, C_N$ that at each time t may fail indep.

$p = \mathbb{P}[C_i \text{ fails at } t]$ so $\mathbb{P}[\exists C_i \text{ that fails at time } t] = 1 - (1-p)^N$

Let X = # time units before the next failure occurs. Due to the fact that each C_i fails independently, we have

$$\mathbb{E}[X] = \frac{1}{1-(1-p)^N} \xrightarrow[N \to \infty]{} 1$$

WORD COUNT WITH 1 ROUND OF MAPREDUCE.

Input: k documents $D_1 \dots D_k \Rightarrow D_i = (\text{name}, \text{list of words})$

D_1 , [orange, apple, pear]

Output $\{(w, c(w))$: w is a word in one of D_i's doc, $c(w) =$ # occurences $\}$ Set of all w followed by $c(w)$

Round 1 → MAP : $\forall D_i$, separately

$D_i \longrightarrow \{(w, C_i(w))$: w is a word in D_i, $C_i(w) =$ # occur. of w in $D_i\}$

→ REDUCE : $\forall w$ separately

L_w = list of $C_i(w)$

$(w, L_w) \to w, \sum_{i=1}^{k} C_i(w)$ with $C_i(w) \in L_w$

ANALYSIS $R = 1$ ⇒ ML ⇒ Suppose that N_i = # of words in D_i

$N_{max} = \max\{D_i, i \in (0,k)\}$

MapPhase : $O(N_{MAX})$ local space

ReducePhase : $O(k)$ local space

$\Rightarrow M_L = O(\max\{N_{max}, k\}) = O(\text{num} + k)$

MA : ⇒ N = # total number of words

Space occupied by input pairs $O(N)$

" intermediate pairs $O(N)$

" output pairs $O(N)$

$\Rightarrow M_A = O(N)$

CLASS COUNT PROBLEM

PROBLEM: Set of N objects o_i, $1 \le i \le N$, labeled with γ_i. We want to count how much objects belong to each class γ.

CLASS COUNT 1 ROUND (NO PARTITIONING) $R=1$

MapPhase: $\forall$ input pairs do $(i, (o_i, \gamma_i)) \to (\gamma_i, 1)$

ReducePhase: Let L_γ = list of $(\gamma_i, 1)$ with $\gamma_i = \gamma$

$$(\gamma, L_\gamma) \to (\gamma, c(\gamma) = |L_\gamma|)$$

Analysis: $R=1 \to M_L \Rightarrow$ RedPhase: if N objects o_i has label γ_i equal, we have

$$\text{MapPhase} = \begin{matrix} O(N) \\ O(1) \end{matrix} \Rightarrow O(N) \text{ doesnt satisfy design goal}$$

CLASS COUNT, 2-ROUND WITH PARTITIONING (deterministic)

Input: $(i, (o_i, \gamma_i))$ with $i \in (0, N($ | ℓ partitions

Round 1 $\forall i$ do $(i, (o_i, \gamma_i)) \to (\underbrace{i \bmod \ell}_{\text{take the rest}}, \gamma_i)$ ← MAPPHASE

Let L_j = List of elements labeled with γ_i, assigned to $i \bmod \ell = j$

$(j, L_j) \to \{(\gamma, c(j,\gamma)) : \forall \text{ label } \gamma \text{ in } j, \text{ count occurences})\}$ ← REDUCE

list of labels with respective # occ. in L_j

Round 2 Empty ← MapPhase

$\forall \gamma$. let L_γ = list of $c(j,\gamma)$'s from outputs of R_1

$$(\gamma, L_\gamma) \to (\gamma, c(\gamma) = \sum_{c(j,\gamma) \in L_\gamma} c(j,\gamma))$$

Analysis:

$R=2 \Rightarrow R_1 \rightsquigarrow$ MapPhase: $O(1)$

RedPhase: $O(N/\ell)$

$\Rightarrow M_L = \max(N/\ell, \ell) \Rightarrow O(\sqrt{N})$

$R_2 \rightsquigarrow$ Map: $\emptyset$

RedPhase: $O(\ell)$

CLASS · COUNT, 2 - ROUND with RANDOM PARTITIONING

ℓ = numb. of part.

Round 1: $\forall$ input $(i, (o_i, \gamma_i)) \rightarrow (x, \gamma_i)$ with x = number $\in [0, \ell)$ with unif. prob.

↓ equal to R_2, R_1 Red Phase of det. partitioning

Analysis: what change is only the MapPhase of R_1: we have that m_x is the number of inter. pairs with key x.

So RedPhase $\Rightarrow O(m)$ where $m = \max\{m_x\}$?

m is surely $\geq \frac{N}{\ell}$ since when partition N objects in ℓ groups there must exists a group with $\frac{N}{\ell}$ objects $\Rightarrow$ hope: $m \geq \frac{N}{\ell}$

Theorem: Fix $\ell = \sqrt{N}$ and suppose that in round 1 keys are assigned to intermediate pairs independently and with uniform prob. from 0 to $\sqrt{N}$. Then, with probability $\geq 1 - \frac{1}{N^5}$ $\Rightarrow m = O(\sqrt{N})$ $\quad [0, \sqrt{N}[$

$$\Downarrow$$

$$M_L = O(\sqrt{N})$$

Proof:

We use 2 things: 1) $\longrightarrow$ Union Bound: $\mathbb{P}\left[\bigcup_{i=1}^{r} E_i\right] \leq \sum_{i=1}^{r} \mathbb{P}[E_i]$ where E_i are events

2) $\longrightarrow$ Chernoff Bound:

$X_1 \dots X_n$ Bernoulli variables s.t. $P[X_i = 1] = p \quad \forall \; i \in \{1, n\}$

So $X = \sum X_i$ is a Binomial(n, p) with $\mathbb{E}[x] = \mu = n \cdot p$

We have that $\forall \; \delta_1 \geq 6$ and $\delta_2 \in (0,1)$:

a) $\mathbb{P}[X \geq \delta_1 \mu] \leq 2^{-\delta_1 \cdot \mu}$

b) $\mathbb{P}[X \leq (1 - \delta_2)\mu] \leq 2^{-\delta_2^2 \mu / 2}$

So the probability is bounded!

so, in R2 MapPhase of W.C algorithm with random partition, we assign a key $x \in [0, \ell[$ with $m_x =$ # number of times an intermediate pairs has key x

Define $y_i := \begin{cases} 1 & \text{if } (\gamma_i, o_i) \to (x, \gamma_i) \\ 0 & \text{otherwise} \end{cases}$

So y_i is a Bernoulli variable with $\mathbb{P}[y_i = 1] = \frac{1}{\sqrt{N}} = p$ $\longrightarrow$ since for hypo $\ell = \sqrt{N}$ and the probability of assignment is uniform $\frac{1}{\ell}$

So $m_x = \sum_{i=0}^{N-1} y_i \Rightarrow m_x$ is a Binomial$\left(N, \frac{1}{\sqrt{N}}\right)$

with $\mathbb{E}[m_x] = N \cdot \frac{1}{\sqrt{N}} = \sqrt{N}$

Using the Chernoff Bound (a) we have $\mathbb{P}[m_x \geq 6\sqrt{N}] \leq 2^{-6\sqrt{N}}$

and for $N \geq 1$ and $\sqrt{N} \geq \log_2 N \Rightarrow 2^{-6\sqrt{N}} \leq 2^{-6\log_2 N} = \left(\left(\frac{1}{2}\right)^{\log_2 N}\right)^6 = \frac{1}{N^6}$

so $\boxed{\mathbb{P}[m_x \geq 6\sqrt{N}] \leq \frac{1}{N^6}}$

Now we want to extend the bound to $m = \max\{m_x : 0 \le x < \sqrt{N}\}$

so we define $E_x =$ "$m_x \ge 6\sqrt{N}$" and so $\mathbb{P}[m \ge 6\sqrt{N}] = \mathbb{P}[E_0 \cup E_1 \cup .. \cup E_{\sqrt{N}-1}]$

$\le \sqrt{N} \cdot \mathbb{P}[E_x] \le \sqrt{N} \cdot \frac{1}{N^6} < \frac{1}{N^5}$ so $\mathbb{P}[m \ge 6\sqrt{N}] \le 1/N^5$ and

by union bound $\sum_{i=1}^{\sqrt{N}-1} P[E_x]$

$P[m < 6\sqrt{N}] \ge 1 - \frac{1}{N^5}$

↓

So, with $P[\cdot] \ge 1 - \frac{1}{N^5}$ we have $m < 6\sqrt{N}$

and then for (N) $m = O(\sqrt{N})$

CONCLUSION: If we distribute N objects at random among l partitions, and N/l is sufficient large, with HIGH PROB. no partition receives an excessive number of objects.

Theorem:

Let S be the set of centers returned by running FFT on P. Then:

$$\phi_{\text{center}}(P,S) \leq 2\,\phi_{\text{center}}(P,k)$$

↪ so the output of FFT is "close" to the optimal solution by a factor of 2

So FFT is a 2-approx. algorithm.

Proof:

$S = \{c_1, c_2, \dots, c_k\}$ is the set of centers selected by FFT, where c_i = i-th center selected } APPROX. CENTERS

Define q = point of P farthest from S, $d(q,S) \geq d(x,S)$

It's important to do an observation: using FFT, c_2 is the farthest point from c_1,

[c_3 is the farthest point from $\{c_2, c_1\}$ $\Rightarrow$ $d(c_2,c_1) \geq d(c_3,c_1) \geq \underbrace{d(c_3,\{c_1,c_2\})}_{\min\{d(c_3,c_1),\, d(c_3,c_2)\}}$

and so, if we iterate, $d(c_k, \{c_1, c_2, \dots, c_{k-1}\}) \geq d(q,S)$]

↳ logic since it would be chosen as a center

Now, consider the set $\{c_1, c_2, \dots, c_k, c_{k+1} = q\} = S \cup \{q\}$

If we prove that $d(q,S) \leq d(c_i, c_j)$ $\forall\, 1 \leq i < j \leq k+1$ the rest of the proof is easy

Fix i,j arbitrarily $1 \leq i < j \leq k+1$ $\Rightarrow$ $d(q,S) \leq d(q, \{c_1, \dots, c_{j-1}\})$

↳ since $j-1 \leq k$ and so being $\{c_1, \dots, c_{j-1}\}$ a subset of S, there are "less possibilities" and we could get at most the result for S

$\leq d(c_j, \{c_1, \dots, c_{j-1}\})$

↳ since FFT takes the next point having as the one at max distance

$\leq d(c_j, c_i)$

↳ since c_j is selected as the max distance and $i<j$, hence $c_i \in \{c_1 \dots c_{j-1}\}$, the $d(c_j, \{c_1 \dots c_{j-1}\})$ is the min distance between c_j and the set and being c_i part of the set, the equivalence holds

Let $S^* = \{c_1^*, c_2^*, \dots, c_k^*\}$ OPTIMAL CENTERS

↳ so $\phi^{OPT}_{kcenter}(P,k) = \phi_{kcenter}(P, S^*) \leq \phi_{kcenter}(P,S)$

- $\forall x \in P,\ d(x, S^*) \leq \phi^{OPT}_{kcenter}(P,k)$ => since $\phi^{OPT}_{kcenter}(P,k) = \max_{x \in P} d(x, S^*)$

For $t = 1..k$ let $C_t^* = \{x \in P : c_t^* \text{ is the center closest to } x\}$ => $P = \bigcup_{i=1}^{k} C_i^*$

We have defined before that the set $\{c_1, c_2, \dots, c_k, c_{k+1} = q\} \subseteq P$

and for the PIGEONHOLE PRINCIPLE must $\exists$ 2 points of $\{c_1 \dots c_{k+1}\}$,

called c_a, c_b that belong to the same cluster C_t^*

=> By triangle inequality => $d(c_a, c_b) \leq d(c_a, c_t^*) + d(c_b, c_t^*) \leq 2\phi^{OPT}_{kcenter}(P,k)$

$= d(c_a, S^*)$

so => $\forall x \in P$ we have that $d(x,S) \leq d(q,S)$ => by the choice of q

$\leq d(c_a, c_b)$ => proof before

$\leq 2\phi^{OPT}_{kcenter}(P,k)$

Lemma: Let T be the union of coresets T_i computed by MR-FFT on P_i

$\forall x \in P$, we have $d(x,T) \leq 2 \cdot \phi^{OPT}_{k\text{center}}(P,k)$ $\Rightarrow$ This implies that T is a good repres. of P, in the sense that each $x \in P$ has a "close-by" representative in T

Proof:

Recall that $P = P_1 \cup P_2 \cup \dots \cup P_l$ and $T = T_1 \cup T_2 \dots \cup T_r$ where $T_i \leftarrow FFT(P_i,k)$

↓ So $\forall$ index $\in (1,l)$ we define q_r = distant point of P_h from T_h $\Rightarrow d(q_r,T_r) \geq d(x,T_r)\ \forall x \in P_h$

CLAIM: $d(q_r,T_r) \leq 2\phi^{OPT}_{k\text{center}}(P,k)\ \forall r$ ~~$\Rightarrow$~~ if it's true means that it's valid $\forall x \in P$ $\Rightarrow$ $d(x,T) \leq d(x,T_r) \leq d(q_r,T_r) \leq 2\phi^{OPT}_{k\text{center}}(P,k)$

↓ due to the fact that $T_r \subseteq T$

↓

We need to prove the claim:

we fix r arbitrarily and we repeat the same argument in the analysis of FFT

$S^* = \{c_1^*, \dots, c_k^*\}$ OPTIMAL CENTERS

C_t^* = optimal clusters around c_t^* with $P = \bigcup_{i=1}^{k} C_t^*$

$T_r = \{c_1 \dots c_k\}$ output of FFT and consider the set $\{c_1, \dots, c_k, c_{k+1} = q_r\}$ that are $k+1$ points $\in P_r$ and so $\in P$

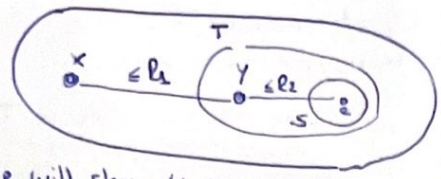
We have that • $d(q_r,T_r) \leq d(c_i,c_j)\ \forall 1 \leq i < j \leq k+1$ previous proof

• $\exists$ 2 points c_a, c_b with $1 \leq a < b \leq k+1$ which $\in$ to some C_t^* for some t

↓ $d(q_r,T_r) \leq d(c_a,c_b) \leq 2\phi^{OPT}_{k\text{center}}(P,k)$

↓ previous proof

Why Composable Coreset technique work well with k-center and FFT



x = arbitrary point of P

y = closest point of T to x

c = closest center of S to y

we will show that $R_1, R_2 \leq 2\,\phi^{OPT}_{kcenter}(P,k)$ $\Rightarrow d(x,S) \leq d(x,c) \leq d(x,y) + d(y,c)$

$\leq R_1 + R_2 \leq 4\,\phi^{OPT}_{kcenter}(P,k)$

ANALYSIS OF MR-FFT

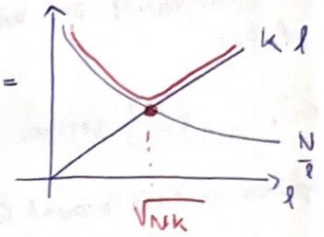
$R = 2$

Round 1: MapPhase $\Rightarrow$ $\forall$ point $p \rightarrow (l, p)$ where l is the number of partitions $\Rightarrow O(1)$

ReducePhase $\Rightarrow O(N/l)$

Round 2: ~~Q~~ $\in$ MapPhase $\Rightarrow M_L = O(\max\{k \cdot l, N/l\}) =$

$O(\sqrt{N \cdot k})$



Theorem: Let S be the set of k centers returned by running MR-FFT on P. Then

$$\phi_{\text{kcenter}}(P,S) \leq 4 \cdot \phi^{OPT}_{\text{kcenter}}(P,k)$$

so obtained

So MR-FFT is a 4-approx algorithm

Proof: From the previous lemma, we know that $\forall x \in P \ \exists y \in T$ s.t.

$$d(x,y) \leq 2\phi^{OPT}_{\text{kcenter}}(P,k)$$

Recall that, as we can see from Pseudocode of MRFFT, S is extracted from T running $FFT(T,k)$.

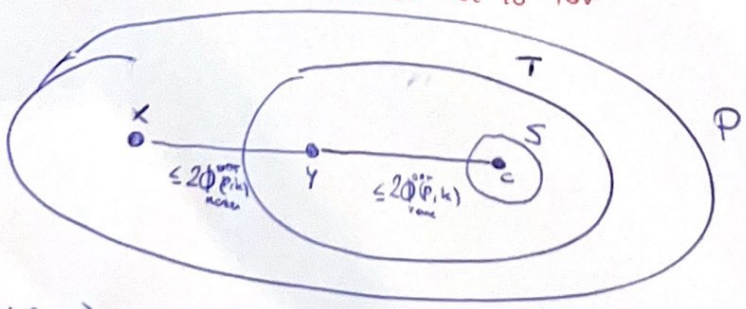
Let $\bar{y}$ the point of T farthest from S and observe that $T \subseteq P$.

The same argument used in the previous lemma with the CLAIM can be repeated here

$$d(\bar{y},S) \leq 2\phi^{OPT}_{\text{kcenter}}(P,k)$$

So how we have that
- $\forall x \in P \ \exists y \in T$ s.t $d(x,y) \leq 2\phi^{OPT}_{\text{kcenter}}(P,k)$ → lemma
- $\forall y \in T, \ d(y,S) \leq d(\bar{y},S) \leq 2\phi^{OPT}_{\text{kcenter}}(P,k)$

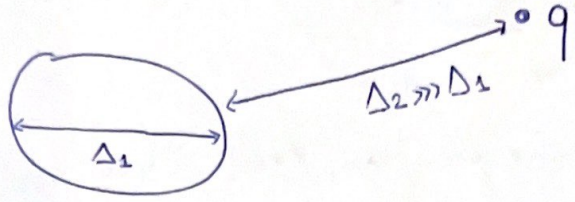
these two thing says that $\forall x \in P \ \exists$ "a good representative" of this point in T and $\forall$ point of T the distance of the closest center is not to far



$$\begin{aligned} d(x,S) &\leq d(x,c) \\ &\leq d(x,y) + d(y,c) \\ &\leq 4\phi^{OPT}_{\text{kcenter}}(P,k) \end{aligned}$$

Example with $k=2$

Suppose that P is done like this:



$N-1$ points

so that points of P are "close" within a cluster and q is an outlier

We have that T is selected randomly from P, with $|T| = \sqrt{N \cdot k}$ and $|P| = N$ with N very large

$$P[q \in T] = \sqrt{N \cdot k} \cdot \frac{1}{N} = \sqrt{\frac{k}{N}} \xrightarrow[N \to +\infty]{} = 0$$

So if T doesn't contain q, and we search for 2 centers in T, for any selection $S \in T$ we have that $\phi(P,S) \gtrsim \Delta_2$ while we know that $\phi_{\text{near means}}(P,k) \approx \Delta_1$

↓ since q is at one cluster

↓ the case where q is a center

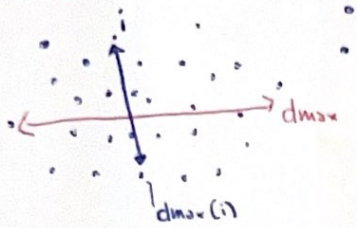
$$\phi(P,S^*) = \max_{x \in P} d(x,S^*)$$

Lemma:

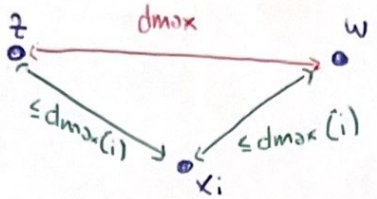
For an arbitrary $x_i \in P$, we define $d_{max}(i) = \max\{d(x_i, x_j) : 0 \leq j < N\}$

For any $0 < i < N$, we have that $d_{max} \in [d_{max}(i), 2d_{max}(i)]$

Proof



- $d_{max} \geq d_{max}(i)$ $\forall i$ => by def of d_{max}
- $d_{max} \leq 2 d_{max}(i)$



By triang. ineq. $d_{max} \leq d(z, x_i) + d(y, x_i) \leq 2 d_{max}(i)$

Is D_t a good approx. of D_{max}?

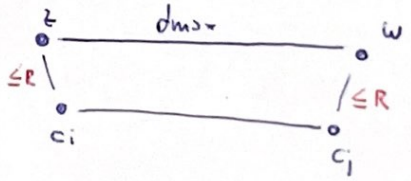
$T = \{c_1, \ldots, c_M\}$, q = point of P farthest from T

$R = d(q, T)$

$d_{max} = d(z, w) \equiv$ true diameter

Let c_i = center of T closest to z

c_j = " to w



$$d_{max} = d(z,w) \leq d(z,c_i) + d(c_i, c_j) + d(c_j, w)$$
$$\leq 2R + d(c_i, c_j) \quad \rightarrow \text{since } R = d(q,T) \geq d(x,T) \ \forall x \in P$$
$$\leq 2R + d_T \quad \rightarrow \text{since } d_T = \max_{(x,y) \in T} d(x,y)$$

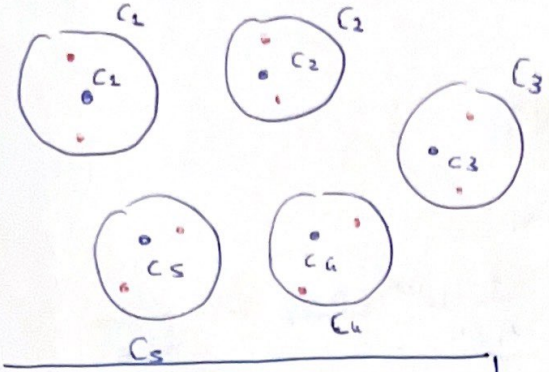
$$\Rightarrow \quad d_T \leq d_{max} \leq d_T + 2R$$

As k grows, the value of R becomes smaller until it's negligible w.r.t d_{max}.

For low d.m. space can be shown that $k \in O(1)$ suffices to obtain $R \leq \epsilon \cdot d_{max}$

Coreset-based algorithm for diversity maximization

Example with $h = 5$, $k = 3$



- RUN FFT to extract T', h centers from P and assign cluster
- Select k points from each cluster

↓

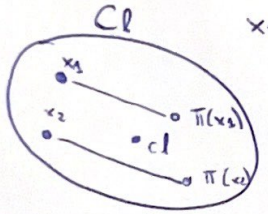
$T = \{c_1 \dots c_5\} \cup \{\text{additional points for each } C_l\}$

Why T is a good coreset?

Let $S^* =$ optimal solution to diversity maximization for P.

IDEA: $\forall$ cluster C_l $(1 \le l \le h)$ create an injective mapping between $S^* \cap C_l$ and $T \cap C_l$ => this is always possible since $|T \cap C_l| = k$ and $|S^* \cap C_l| \le |S^*| = k$

For example



$x_1, x_2 \in S^*$

$\pi(x_1), \pi(x_2) \in T \cap C_l$

$\pi(x_i) =$ PROXY OF x_i

Let R be the radius of the clustering i.e. $d(x, \{c_1 \dots c_h\}) \le R \quad \forall x \in P$

For $i = 1, 2$

$$d(x_i, \pi(x_i)) \le d(x_i, c_l) + d(c_l, \pi(x_i)) \le 2R$$

=> Means that point of S^* has a "nearby" proxy in T!

We now agree that T contains a good solution to div. maxim. for P

Consider the set $S = \{\pi(x) : x \in S^*\} \subset T \quad |S| = k$

$$div(S) = \sum_{x_i, x_j \in S^*} d(\pi(x_i), \pi(x_j)) \ge \sum_{x_i, x_j \in S^*} d(x_i, x_j) - d(x_i, \pi(x_i)) - d(x_j, \pi(x_j)) \ge \sum_{\substack{x_i, x_j \\ i<j \\ \in S^*}} (d(x_i, x_j) - 4R)$$

↓ triangle inequality

$$= div^{OPT}(P, k) - \binom{k}{2} 4R$$

↓ (since it's done $\forall$ couple of points and $\forall$ combinations)

If h is large enough we can make $4R \leq \epsilon \, \Phi^{OPT}_{kcenter}(P,k)$ - by fact

In this case $div(S) \geq div^{OPT}(P,k) - 4R\binom{k}{2}$

$$\geq div^{OPT}(P,k) - \epsilon \, \Phi^{OPT}_{kcenter}(P,k)\binom{k}{2}$$

$$\geq div^{OPT}(P,k) - \epsilon \, div^{OPT}(P,k)$$

$$\Rightarrow div(S) \geq (1-\epsilon)\, div^{OPT}(P,k) = \frac{1}{1+\epsilon'} div^{OPT}(P,k)$$ with ϵ' close to ϵ

$$div^{OPT}(T,k) \geq div(S) \geq \frac{1}{1+\epsilon'} div^{OPT}(P,k) \Rightarrow \checkmark$$

Example of application of Boyer-Moore algorithm

$\Sigma = x_1 x_2 \dots x_9$ ⇒

A A A C C B B A A

step	Cand	Count
0	NULL	0
1	A	1
2	A	2
3	A	3
4	A	2
5	A	1
6	A	0
7	B	1
8	B	0
9	A	1

⇒ at the end: cand = A ~ true majority

Theorem: Given a stream Σ with a majority element m, the Boyer Moore algorithm returns m using:

- working memory of size $O(1)$
- 1 pass
- $O(1)$ time per element

Proof:

- working memory $O(1)$
- 1 pass
- $O(1)$ time

} straightforward (simple and direct proof)

CORRECTNESS: we must show that if a majority elements **m** exists, then at the end of the stream **cand = m**

For $t = [0, n]$, with $n = |\Sigma|$, let $cand_t$, $count_t$ be the values of cand and count after $x_1 \dots x_t$ ($t=0$ means initialization value)

It's easy to show that after processing $x_1 \dots x_t$, we have that these t elements can be partitioned into:
- count occurences of cand
- $(t - count)/2$ pairs (e_1, e_2) with $e_1 \neq e_2$

For the above example, $t = 5$ ⇒ $x_1 \dots x_t$ = A A A C C ~ cand = A, cont = 1

⇒ 1 occurrence of A and 2 pairs (A, C), (A, C)

We now show that $cand_{|\Sigma|} = m$.

By contradiction suppose that $cand_n \neq m$. Because of the invariants stated before, we have that Σ contains

- $count_n \geq 0$ occurrences of $cand_n$
- $(n - count_n)/2$ pairs (e_1, e_2) with $e_1 \neq e_2$

Therefore, if $m \neq cand_n$, it can occur at most $(n - count_n)/2 \leq \frac{n}{2}$ since

and not $\frac{n - ncount}{2} + ncount$, It can appear only in couples (e_1, e_2)

$count_n \geq 0 \Rightarrow$ m cannot be the majority $\Rightarrow$ contradiction

Theorem:

Let $\Sigma = x_1, x_2 \ldots$ For any time $t \geq m$, the set S maintained by the reservoir sampling algorithm is an m-sample of $\Sigma_t = x_1 \ldots x_t$

Proof:

Let $S_t = S$ after processing $x_1 \ldots x_t$. We now show by induction on $t \geq m$ that S_t is an m-sample of $x_1 \ldots x_t$, that is, for each $i \in [1, t]$, we must have that $\mathbb{P}[x_i \in S_t] = m/t$

Base of the induction: $t = m$, we can choose only these elements ✓

Inductive step: Suppose that the property holds for $t-1 \geq m$

so $\forall i \in [1, t-1] \Rightarrow \mathbb{P}[x_i \in S_{t-1}] = \frac{m}{t-1}$

Now we evaluate $\mathbb{P}[x_i \in S_t]$ $\forall i \in [1, t]$:

- $\mathbb{P}[x_t \in S_t] = \frac{m}{t}$ by construction of the algorithm
- Consider x_i, $i < t$, and define the event $A =$ "x_t is added to S at time t" ~

↳ we have that $\mathbb{P}[A] = \frac{m}{t}$ $\quad \mathbb{P}[\sim A] = 1 - \frac{m}{t}$

(since the algorithm chooses with prob $\frac{m}{t}$ if evict an x_i in favore of x_t)

By the law of total probabilities

$$\mathbb{P}[x_i \in S_t] = \mathbb{P}[x_i \in S_t \wedge A] + \mathbb{P}[x_i \in S_t \wedge \sim A]$$

$$= \mathbb{P}[x_i \in S_t \mid A] \cdot \mathbb{P}[A] \sim \frac{m}{t}$$

$$= \mathbb{P}[x_i \in S_{t-1} \mid A] \cdot \mathbb{P}[x_i \text{ not evicted in step } t \mid A]$$

$\mathbb{P}[x_i \in S_{t-1} \mid A] = \frac{m}{(t-1)}$ $\quad \mathbb{P}[x_i \text{ not evicted in step } t \mid A] = 1 - \frac{1}{m}$

$$\left[\frac{m}{t-1} \cdot \left(1 - \frac{1}{m}\right) \cdot \frac{m}{t}\right] \rightarrow$$

$$\mathbb{P}[x_i \in S_t \wedge \sim A] = \mathbb{P}[x_i \in S_t \mid \sim A] \cdot \mathbb{P}[\sim A]$$

$\mathbb{P}[\sim A] = 1 - \frac{m}{t}$

$\mathbb{P}[x_i \in S_{t-1}] = \frac{m}{(t-1)}$

$$\left[\frac{m}{t-1} \cdot \left(1 - \frac{m}{t}\right)\right]$$

$+ =$

$$\boxed{\frac{m}{t}}$$

Theorem: Stickly sampling solves the ϵ-AFI problem correctly with $\mathbb{P}[\cdot] \geq 1-\delta$ and requires

- working memory of size $O(\ln(1/\delta\varphi)/\epsilon)$ in expectation (r)
- 1 pass
- $O(1)$ expected time per element

it returns: NOT RET | FALSE POSITIVES | RETURNED ✓ ($\varphi-\epsilon$, φ)

Proof: Recall that $\Sigma = x_1 \dots x_n$

so 2,3) are straightforward.

2) each input is seen only one time, so at the end the frequency of items 1 pass

3) whence we have find the item (working memory) we need to increment a count

WORKING MEMORY:

- It's proportional to the size of S and, if S is a properly set, the size of S is proportional to the number of items it contains
- Each x_i contributes an extra unit of space to S with prob $\leq \frac{r}{n}$ since it's added with probability $\frac{r}{n}$ only if not already in S

$\Rightarrow$ Expected working memory $O(n \cdot \frac{r}{n}) = O(r)$ and $r = \frac{\ln\left(\frac{1}{\delta\varphi}\right)}{\epsilon}$

(n items · $\frac{r}{n}$ prob)

CORRECTNESS: We need to show that with $\mathbb{P}[\cdot] \geq 1-\delta$, the set of items returned is s.t.:

A) all true freq. items are in the set

B) no item occurring $< (\varphi-\epsilon)\cdot n$ times in Σ are in the set

↳ ensured by the fact that we return only items whose # of occurences is $\geq (\varphi-\epsilon)\cdot n$

↳ Let's verify A)

Consider a frequent item a $\Rightarrow$ $\mathbb{P}[a \text{ is not returned in the output}] \leq \mathbb{P}[\text{none of the first } \lceil \epsilon \cdot n \rceil \text{ occurren. of } a \text{ in } \Sigma \text{ are sampled}]$ $\Rightarrow$

$$\Rightarrow = \left(1-\frac{r}{n}\right)^{\lceil \epsilon \cdot n \rceil} \leq \left(1-\frac{r}{n}\right)^{\epsilon \cdot n} = \left(1-\frac{r}{n}\right)^{\frac{n}{n}\cdot\frac{r}{r}\cdot\epsilon} = \left(\left(1-\frac{1}{n/r}\right)^{n/r}\right)^{\epsilon \cdot r} \leq \left(\frac{1}{e}\right)^{\epsilon \cdot r}$$

where we use the fact that $\left(1-\frac{1}{x}\right)^x \leq \frac{1}{e} \quad \forall x \geq 1$

Why? because S returns $\geq (\varphi-\epsilon)\cdot n$ items. So, if $\lceil \epsilon \cdot n \rceil$ occ. are missed, we have $\leq \varphi$ occ. $\Rightarrow$ not ret.

Let $a_1 \dots a_k$ be the (true) freq. items and recall that $k \leq \frac{1}{\varphi}$

$$\mathbb{P}[\text{some } a_i \text{ not returned}] \leq \sum_{i=1}^{k} \mathbb{P}[a_i \text{ not returned}] \leq k \cdot \left(\frac{1}{e}\right)^{\epsilon \cdot r} \leq$$

(U.B)

$$\leq \frac{1}{\varphi}\left(\frac{1}{e}\right)^{\epsilon \cdot \ln(\cdot\cdot)/\epsilon} = \frac{1}{\varphi}\left(\frac{1}{e}\right)^{\ln\left(\frac{1}{\delta\varphi}\right)} = \delta \Rightarrow \boxed{\mathbb{P}[\text{all items are returned}] \geq 1-\delta}$$

Example

U = alphabet $|U| = 26$ => we need to represent $L = 25$ in binary: 5 bits!

Σ = A, D, A, A, C, B, F, F, B, A, E, C

x	h(x)	tr(h(x))	C (6 bits)
A	01001	0	1 0 0 0 0 0
D	11100	2	1 0 1 0 0 0
C	11000	3	1 0 1 1 0 0
B	01110	1	1 1 1 1 0 0
F	10100	2	"
E	10011	0	"

$R = 3 \Rightarrow \tilde{F_0} = 2^{\ell} - 3$ while $F_0 = 6$

Why does it work? (intuition)

For semplicity, assume $|U|$ a power of 2

For $x \in \Sigma$, what is the prob that $h(x)$ has at least j trailing zeros?

$$P[tr(h(x)) \geq 1] = \frac{1}{2}$$

$$\vdots$$

$$P[tr(h(x)) \geq j] = \frac{1}{2^j}$$

Motivation: interval $[0, |U|-1]$ contains $\frac{|U|}{2^j}$ integers whose binary configuration is

E.g. ... 0 0 0 .. 0 (with j trailing zeros)

2^{ℓ} configurations with j trailing zeros. So, a random integer have $\geq \frac{1}{2^j}$ probability to be an integer with j's zero (trailing)

The algorithm will generate F_0 (=> F_0 since distinct element) random integers in $[0, |U|-1]$ and the expected number of these integers with $\geq j$ trailing zeros is $\frac{F_0}{2^j}$ => due to indep. $B(F_0, p)$ => $E[X] = F_0 \cdot p \geq \frac{F_0}{2^j}$

So we expect to generate integers with $\geq j$ trailing zero's only if $\frac{F_0}{2^j} \geq 1$

Therefore, the maximum numbers of trailing 0's obtained when processing the stream Σ (i.e. ℓ) is expected to be s.t

$\frac{F_0}{2^{\ell}} \geq 1$ but $\frac{F_0}{2^{\ell+1}} < 1$ => $2^{\ell} \approx F_0$

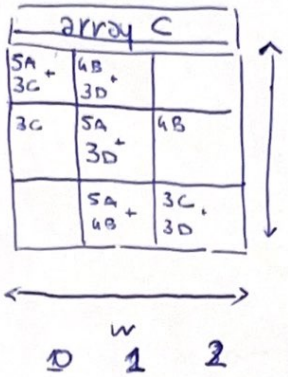
(ℓ because of one element with ℓ trailing 0 with $\ell+1$)

example:

$n = 15$, $d = 3$, $w = 3$

$\Sigma = A, B, C, B, D, A, C, D, A, B, D, C, A, A, B$

u, fu	h_0	h_1	h_2
A, 5	0	1	1
B, 4	1	2	1
C, 3	0	0	2
D, 3	1	1	2



$\tilde{f}_A = \min\{8, 8, 9\} = 8 > f_A = 5$

$\tilde{f}_B = \min\{7, 6, 9\} = 6 = f_B$

$\tilde{f}_C = \min\{8, 3, 6\} = 3 = f_C$

$\tilde{f}_D = \min\{9, 8, 6\} = 6 > f_D = 3$

Theorem: Consider a $d \times w$ count min sketch for a stream Σ of length n, where $d = \log_2\left(\frac{1}{\delta}\right)$ and $w = \frac{2}{\epsilon}$, for some $\epsilon, \delta \in (0,1)$. The sketch ensures that, for any given $u \in U$, occurring in Σ,

$$\tilde{f_u} - f_u \leq \epsilon \cdot n$$

with prob $1 - \delta$

because considering a precise column and row, there is $\frac{1}{w}$ prob that each item goes there.

$\mathbb{E}[\cdot] = \sum u \cdot \frac{1}{w} = \frac{\sum u}{w} \Rightarrow \frac{n}{w}$

Proof: Fix an arbitrary $u \in U$ and $j \in [0, d-1]$

- $C[j, h_j[u]]$ receives, in expectation, a fraction w of the total mass $n = |\Sigma|$
  - => receives a value $\frac{n}{w}$ and includes for sure f_u => $\mathbb{E}[C[j, h_j(u)] - f_u] \leq \frac{n}{w} = \frac{\epsilon n}{2}$

since we insert in the table surely fu times u

- By Markov inequality

$$\mathbb{P}[C[j, h_j(u)] - f_u \leq \epsilon \cdot n] \geq \frac{1}{2} \Rightarrow \mathbb{P}[C[j, h_j(u)] - f_u > \epsilon \cdot n] < \frac{1}{2}$$

Since $\tilde{f_u}$ is the minimum among those rows ($d = \log_2\left(\frac{1}{\delta}\right)$), we can say that $\tilde{f_u} - f_u > \epsilon \cdot n$ only if all rows $> \epsilon \cdot n$ and this happen with prob $< \left(\frac{1}{2}\right)^d = \delta$

So $\mathbb{P}[\tilde{f_u} - f_u \leq \epsilon \cdot n] \geq 1 - \delta$

example:

$h=15$, $d=3$, $w=3$

$\bar{S} = A, B, C, B, D, A, C, D, A, B, D, C, A, A, B$

u, f_u	h_0	g_0	h_1	g_1	h_2	g_2
A, 5	0	1	0	1	1	1
B, 4	1	-1	2	1	1	-1
C, 3	0	-1	0	-1	2	1
D, 3	1	-1	1	1	2	1

array C

$5A-3C$	$-4B-3D$	
$-3C$	$5A+3D$	$4B$
	$5A-4B$	$3C+3D$

$\tilde{f}_{a,0}=2 \quad \tilde{f}_{a,1}=8 \quad \tilde{f}_{a,2}=1 \Rightarrow \tilde{f}_a = \text{median}\{2,8,1\} = 2 < f_a = 5$

$\tilde{f}_B = \text{median}\{7,4,-1\} = 4 = f_B$

$\tilde{f}_C = \text{median}\{-2,3,6\} = 3 = f_C$

$\tilde{f}_D = \text{median}\{7,8,6\} = 7 > f_D$

Theorem

Proof: We prove only (A)

Fix $u \in U$ and $j \in [0, d-1]$ arbitrarily. We show that $\mathbb{E}[\tilde{f}_{u,j}] = f_u$

↳ $g_j(u) \cdot C[j, h_j(u)]$

$\forall a \in U, a \neq u$, define

$$Y_a = \begin{cases} f_a & \text{if } h_j(a) = h_j(u) \text{ AND } g_j(a) = g_j(u) \\ -f_a & \text{if } h_j(a) = h_j(u) \text{ AND } g_j(a) \neq g_j(u) \\ 0 & \text{i.e } h_j(a) \neq h_j(u) \end{cases}$$

⇒ are in the same cell for row i and have the same sign

CRUCIAL OBSERVATION: Y_a = contribution of $a \neq u$ to $\tilde{f}_{u,j}$

$$\tilde{f}_{u,j} = f_u + \sum_{a \in U, a \neq u} Y_a$$

Now for $a \neq u$

$P[Y_a = f_a] = P[Y_a = -f_a] = \frac{1}{w} \cdot \frac{1}{2}$ ← $g_j(a)$ random $\in \{-1, +1\}$

$P[Y_a = 0] = 1 - 2 \cdot \frac{1}{w} \cdot \frac{1}{2} = 1 - \frac{1}{w}$ ← $h_j(a) = h_j(u)$

↓ 1 - 2 possibilities above

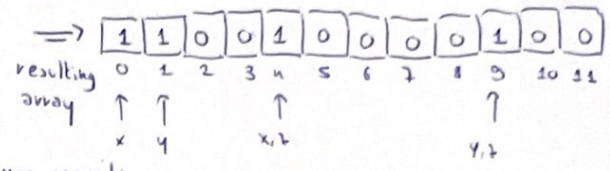
$$\mathbb{E}[\tilde{f}_{v,j}] = \mathbb{E}\left[f_v + \sum_{a \in U, v \neq a} y_a\right] = f_v + \mathbb{E}[\Sigma y_a] = f_v + \sum \mathbb{E}[y_a] = f_v$$

$$f_a\left(\frac{1}{\omega}, \frac{1}{2}\right) - f_a\left(\frac{1}{\omega}, \frac{1}{2}\right) = 0$$

example

$S = \{x, y, z\}$ $n = 12$ $k = 2$

	x	y	z
h_0	0	1	4
h_1	4	9	5



Check(x): $h_0(x) = 0$ $h_1(x) = 4$ Yes, true positive

Check(T): $h_0(T) = 4$ $h_1(T) = 7$ No, true negative

not = 1

Check(P): $h_0(P) = 1$ $h_1(P) = 9$ Yes, ~~true~~ positive
False]

Theorem:

Proof: Let $S = \{e_1 .. e_m\}$

Under the assumption that the k hash function are independent, provide indeces uniformaly in $[0, n-1]$ and $h_j(e')$ and $h_j(e'')$ are independent, $\forall j \in [0, k-1]$ and $k \in e' \in e''$

The indices of the cells of A which contain 1's can be seen as $k \cdot m$ indep. variable uniformely distributed in $[0, n-1]$

k hash function, |S|

=> for any index $l \in [0, n-1]$, we have that

$$\mathbb{P}[A[l] = 0] = \left(1 - \frac{1}{n}\right)^{k \cdot m} = \left(1 - \frac{1}{n}\right)^{n \cdot \frac{km}{n}} = \left(\left(1 - \frac{1}{n}\right)^{n}\right)^{\frac{km}{n}} \approx \left(\frac{1}{e}\right)^{\frac{km}{n}}$$

true if n is large enough

we define $p = e^{-\frac{km}{n}}$ and we make the SIMPLYFING ASSUMPTION that A contains exactly $p \cdot n$ 0's (the theorem says true ever without this assumption)

Consider $x \in \Sigma$ s.t $x \notin S$ and let $l_j = h_j(x)$ $0 \le j < k$

Since $h_j(x)$ is uniformely distubuted in $[0, n-1]$

$$\mathbb{P}[A[l_j] = 1] = 1 - \underbrace{\frac{pn}{n}}_{\mathbb{P}[A[l_j] = 0]} = 1 - p$$ i.e prob of not hitting a 0

$$\mathbb{P}[A[l_j] = 1 \ \forall 0 \le j < k] = \prod_{j=0}^{k-1} \mathbb{P}[A[l_j] = 1] = (1-p)^k = \left(1 - e^{-\frac{km}{n}}\right)^k$$

Proof:

* PROBABILISTIC CORRECTNESS

Case 1: $B_r(q) \cap P \neq \emptyset$. In this case, a legal output is any point $p' \in P$ s.t.

$$d(q, p') \leq c \cdot r$$

Now consider an arbitrary point $p \in B_r(q) \cap P$ (exist)

Then: $P_r(\text{answer correct}) = P_r[\text{answer} \neq \text{null}] \geq P[p \text{ is mapped in the same region by } h]$

$= p_1$

by property of H

Case 2: $B_r(q) \cap P = \emptyset$. The answer will be surely correct both if a point p at distance $\leq c \cdot r$ is returned and if null is RETURNED.

* CONSTRUCTION AND SPACE TIME: Straightforward

* QUERY TIME: Suppose that the bucket $T[h(q)]$ is implemented as a list.

The scan of $T[h(q)]$ ends as soon as a point at distance at most $\leq c \cdot r$ from q is found or the end of the list is reached. Therefore, at most $x+1$ points of $T[h(q)]$ are checked where x is the number of points in the bucket at distance $> c \cdot r$ from q.

Since a far point (i.e at distance $> c \cdot r$) has prob. at most p_2 of being in the bucket $T[h(q)]$ and since there are at most n far points in P (all the points), Then, the expected number $\mathbb{E}[x] \leq n \cdot p_2$