Weak vs. Strong Learning

Def. Algorithm $\mathcal{A}$ weakly "PAC learns" concept class $\mathcal{C}$

if $\forall\, c \in \mathcal{C}$ & $\forall$ dists $\mathcal{D}$ $\quad \exists\, \gamma > 0$

$\forall\, \cancel{\varepsilon}, \delta > 0$ $\quad$ ($\delta = \frac{1}{4}$ or $\frac{1}{n^2}$ doesn't affect)

with prob $\geq 1-\delta$

given examples of c

$\mathcal{A}$ outputs h s.t. $\Pr_{\mathcal{D}}[h(x) \neq c(x)] \leq \cancel{\varepsilon}\ \frac{1}{2} - \frac{\gamma}{2}$

$\uparrow$ advantage

It was conjectured that distribution free weak learning was really weaker but surprise!

Can "boost" a weak learner

Thm if $\mathcal{C}$ can be weakly learned on <u>any</u> dist $\mathcal{D}$ then $\mathcal{C}$ can be (strongly) learned. $\Leftarrow$ i.e. $\forall\, \varepsilon, \mathcal{D}_0$

Will prove for case of $\mathcal{D}_0 = \mathcal{U}$

Applications

1) "Theoretical"

- Unif dist Algorithms for poly term DNF
  weight w - poly threshhold fctns

  } low degree alg doesn't work wel

  ∴ (Boosting + KM)

- Ave case vs. worst case complexity

2) practical - Boosting

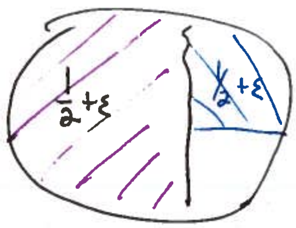
   Freund - Schapire

Good & Bad Ideas

1) simulate weak learner several times on same distribution & take majority answer
   -or-
   best answer

   gives better confidence
   but doesn't reduce error, what if always get same answer?

2) filter out examples on which current hypothesis does well & run weak learner on part where you do badly.



Problem: given a new example, how do you know which section it is in?

3) Keep some samples on which you are ok in filtering
always use majority vote on all previous hypotheses
to predict value of new samples

history : Schapire, Freund-Schapire, Impagliazzo–Servedio. Klivans

Filtering Procedures

- decide which samples to keep, which to throw out
- samples on which so far you guess correctly ← need for check future hypot
  incorrectly ← need to improve on these

The setting

- Given labelleled examples

$$(x_1, f(x_1)), (x_2, f(x_2)), \ldots$$

$$x_i \in_R \mathcal{D}$$
$$f \in \mathcal{C}$$

- Given weak learning alg WL which weakly learns (advantage $\frac{\gamma}{2}$) on any dist $\mathcal{D}'$

Boosting Algorithm

- Stage 0 (Initialize)

$\mathcal{D}_0 \leftarrow \mathcal{D}$

run WL on $\mathcal{D}_0$ to generate (whp)

C_1 s.t. $\Pr_{\mathcal{D}_0}[f(x) = C_1(x)] \geq \frac{1}{2} + \gamma/2$

- For $i = 1 \ldots T = O(\frac{1}{\gamma^2 \varepsilon})$ stages, stage i: (can stop if Majority($C_1 \ldots C_i$) correc on $\geq 1-\varepsilon$ inp

(1) Construct $\mathcal{D}_i$ via "filtering procedure":

{ favor pts on which maj of $C_1 \cdots C_i$ don't do well
but also keep some other points }

Will specify soon

(2) run WL on examples from $\mathcal{D}_i$ to output

C_{i+1} s.t. $\Pr_{\mathcal{D}_i}[f(x) = C_{i+1}(x)] \geq \frac{1}{2} + \frac{\gamma}{2}$

- output $C = MAJ(C_1 \ldots C_T)$

Filtering procedure

Given new example $x, f(x)$ from example oracle

- if majority of $c_1 \dots c_i$ wrong, keep it

  ie $\geq \frac{i}{2}$

- if large majority right, then discard

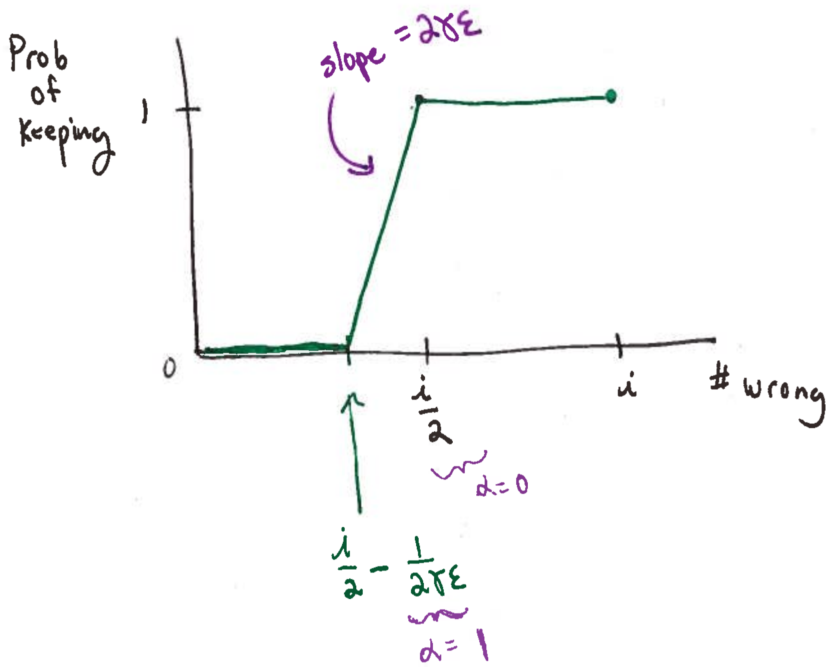
  ie. $\#\text{right} - \#\text{wrong} > \frac{1}{\gamma\varepsilon}$

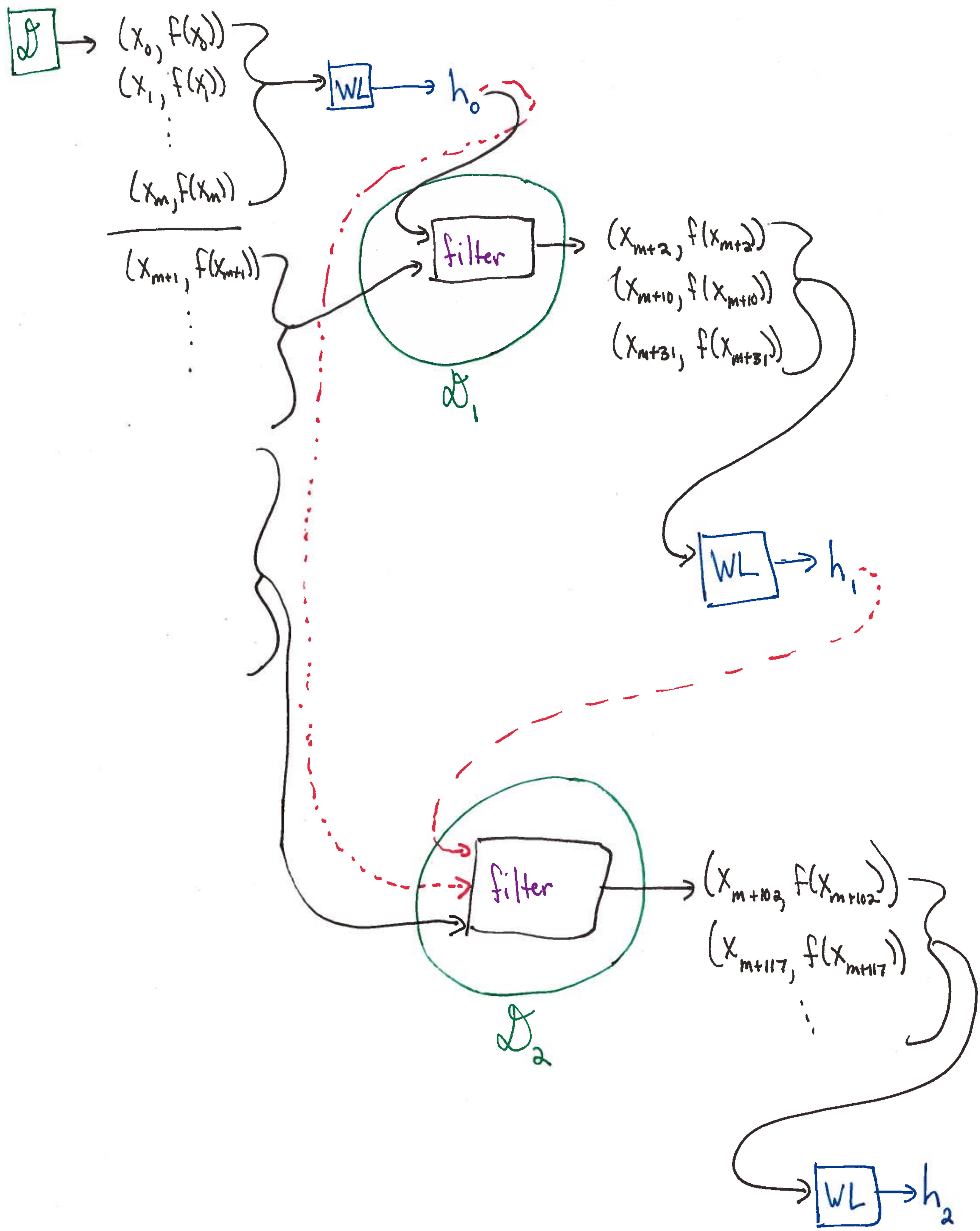
  or $\#\text{wrong} \leq \frac{i}{2} - \frac{1}{2\gamma\varepsilon}$

- else $\#\text{right} - \#\text{wrong} = \frac{\alpha}{\gamma\varepsilon}$ for $0 < \alpha < 1$

  $\#\text{wrong} - \#\text{right} = \frac{-\alpha}{\gamma\varepsilon}$

  so keep with prob $= 1 - \alpha$




$\mathcal{D}$
$(x_0, f(x_0))$
$(x_1, f(x_1))$
$(x_m, f(x_m))$
WL
h_0
$(x_{m+1}, f(x_{m+1}))$
filter
$\mathcal{D}_1$
$(x_{m+2}, f(x_{m+2}))$
$(x_{m+10}, f(x_{m+10}))$
$(x_{m+31}, f(x_{m+31}))$
WL
h_1
filter
$\mathcal{D}_2$
$(x_{m+102}, f(x_{m+102}))$
$(x_{m+117}, f(x_{m+117}))$
WL
h_2

Need to show:

1) Output ~~is~~ has nontrivial agreement with f

2) # samples needed not too bad

why could it be bad?

if throw out lots of samples, might need to wait a long time before WL can give an output.

but if throw out too many samples then you already have a good hypothesis!

will stop if $Maj(c_1 \ldots c_i)$ correct on $\geq 1-\varepsilon$ fraction of inputs

o.w. $Maj(c_1 \ldots c_i)$ incorrect on $> \varepsilon$ fraction

so filtering procedure outputs sample with prob $\geq \varepsilon$ (+ in expectation, every $1/\varepsilon$ samples of $\mathcal{D}$ at least one makes it thru the filtering system)

Thm filter outputs new sample in time $\leq \frac{1}{\text{error}[Maj(c_1 \ldots c_i)]}$

$\Rightarrow$ filtering slows down sample collection by $\leq O(1/\varepsilon)$

So lets focus on 1

Notation

- $R_c(x) = \begin{cases} +1 & \text{if } f(x) = c(x) \\ -1 & \text{if } f(x) \neq c(x) \end{cases}$ "is c correct on x?"

- $N_i(x) = \sum_{1 \le j \le i} R_{c_j}(x)$ after iteration i, how many c's correct? (#right − #wrong)

- $M_i(x) = \begin{cases} 1 & \text{if } N_i(x) \le 0 \\ 0 & \text{if } N_i(x) \ge \frac{1}{\varepsilon\gamma} \\ 1 - \varepsilon \cdot \gamma \cdot N_i(x) & \text{o.w.} \end{cases}$ prob of keeping x in filtering (after stage i)

note — all "wrong" x included in M
also some "right" x included

Note that new distribution on samples is proportional to M_i:

- $D_{M_i}(x) \equiv \frac{M_i(x) D_0(x)}{\sum_y M_i(y) D_0(y)}$ since D_0 is assumed to be uniform, we can drop this & get $\frac{M_i(x)}{\sum M_i(y)}$ distribution induced by M

note $D_{M_i}(x) = \mathcal{D}_i$

$\sum_y M_i(y)$ includes all "wrong" y but also y for which may that isn't overwhelming are correct } upper bounds # wrong y

How correct are we wrt. D_{M_i}?

- $Adv_c(M_i) = \sum_x R_c(x) M_i(x)$ "Advantage" of c on M
$\sim Pr[\text{correct}] - Pr[\text{incorrect}]$
$= 2 \cdot Pr[\text{correct}] - 1$

- $Pr_{x \in D_{M_i}}[c(x) = f(x)] = \frac{1}{2} + \frac{Adv_c(M_i)}{2 \cdot \sum_x M_i(x)}$

Note:

if $\sum M_i(x) \geq \varepsilon\, 2^n$

$$Adv_c(M_i) \geq \gamma \cdot \varepsilon \cdot 2^n$$

convert claim about WL $\Rightarrow$ claim about advantage

ie. if have γ advantage on output of 1

& still almost wrong on lots of inputs

then new advantage is pretty good

if not, then you are done

Begin Proof

For input x

let $A_i(x) \leftarrow \sum_{0 \leq j \leq i-1} R_{c_{j+1}}(x)\, M_j(x)$

strange – indices don't ma

$c_1 \cdots c_j$ define D_j

but c_{j+1} learned fr

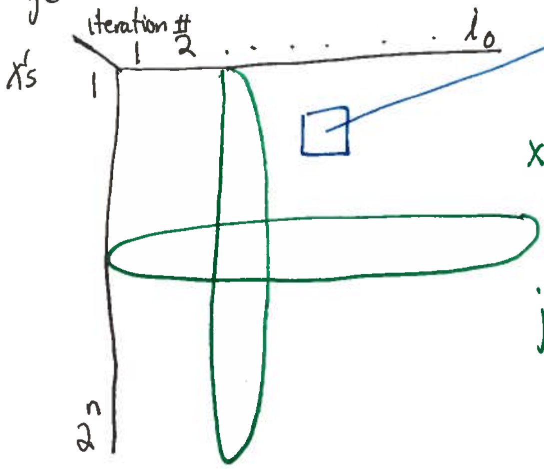
WL on D_j

Claim $A_i(x) \leq \frac{1}{\varepsilon\gamma} + \frac{\varepsilon\gamma}{2} \cdot i$

- bounds advantage per input
- only helps after $\frac{1}{\varepsilon\gamma}$ rounds

Plan for use of claim:

Consider large matrix:



$(k,j)^{th}$ Entry: $\underbrace{R_{c_{j+1}}(x_k)\, M_j(x_k)}_{\text{weighted version of whether } c_{j+1} \text{ correct on } x}$

x's row sum $= \sum_{0 \leq j \leq i_0} R_{c_{j+1}}(x)\, M_j(x) = A_{i_0+1}(x)$

j^{th} col sum $= \sum_x R_{c_{j+1}}(x)\, M_j(x)$

$= Adv_{c_{j+1}}(M_j) \leftarrow \geq \gamma \cdot \sum_x M$

else algorit

Goal: lower/upper bound average entry in matrix

lower bound:

lower bound average entry in column via

- correctness of WL
- fact that algorithm proceeds
  $\Rightarrow$ lots of error
  $\Rightarrow \sum_x M_j(x)$ big
  $\Rightarrow$ lots of progress in WL in absolute terms

upper bound:

upper bound rows via claim

- if advantage is too much, lose measure
  this is where majority rule & weighting scheme is used

More details:

Assume claim, prove theorem:

Assume haven't terminated at i_0+1^{th} stage

- so $\text{error}(C_{i_0}) \geq \varepsilon$

$$\sum_x M_{i_0}(x) \geq \varepsilon 2^n$$

Claim $\Rightarrow$

$$\sum_x A_{i_0+1}(x) = \sum_x \sum_{0 \le j \le i_0} R_{c_{j+1}}(x)\, M_j(x) \qquad \text{def of } A_{i_0+1}$$

$$= \sum_{0 \le j \le i_0} Adv_{C_{j+1}}(M_j) \qquad \text{def of } Adv_{C_{j+1}}$$

$$\geq (\underbrace{\gamma \varepsilon 2^n}_{\text{from "note"}})(i_0+1)$$

$$+ \quad \sum_x A_{i_0+1}(x) \le \sum_x \left(\frac{1}{\varepsilon\gamma} + \frac{\varepsilon\gamma}{2}\cdot(i_0+1)\right) \qquad \text{claim}$$

$$= 2^n\left(\frac{1}{\varepsilon\gamma} + \frac{\varepsilon\gamma}{2}(i_0+1)\right)$$

putting together:

$$(\varepsilon\gamma)(i_0+1) \le \frac{1}{\varepsilon\gamma} + \frac{\varepsilon\gamma}{2}(i_0+1)$$

so $$\frac{\varepsilon\gamma}{2}(i_0+1) \le \frac{1}{\varepsilon\gamma} \Rightarrow i_0 \le \frac{2}{\varepsilon^2\gamma^2} - 1$$

$\blacksquare$

Proof of claim:

Question: how can an input x add to cumulative advantage througout algorithm?

Observations:

- if algorithm's hypotheses $c_1 \dots c_i$ are overwhelmingly correct on x, then not at all because x gets measure 0
- if algorithm's hypotheses are doing badly (mostly wrong) then not at all because they decrease advantage
- Main Issue:

  can wander in middle - $\underbrace{\text{majority}}_{\text{increase advantage}}$ correct but $\underbrace{\text{not large majority}}_{\text{so have positive measure}}$

need to bound this case.

Proof of Claim

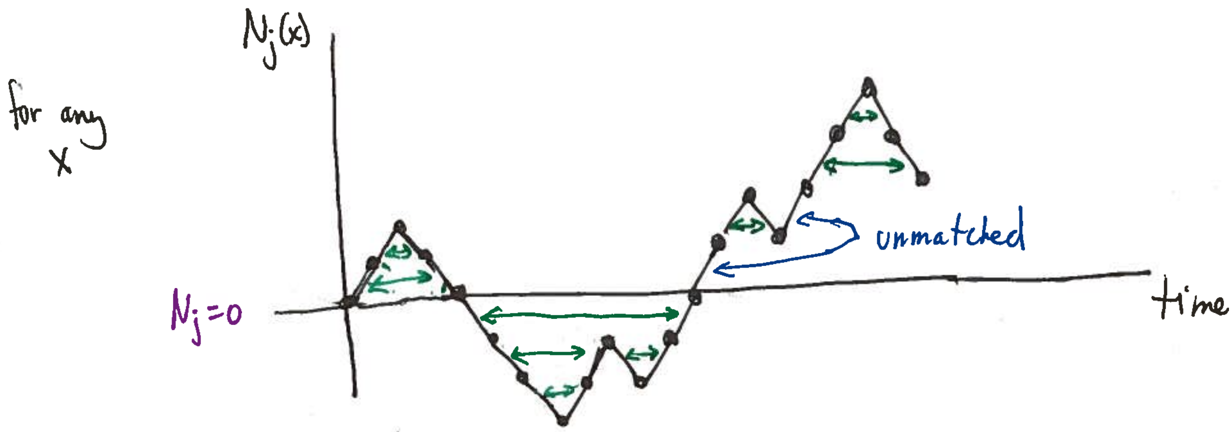
First, strange but obvious fact:

Fact "elevator argument"

If one spends any amount of time in an elevator, then no matter what sequence of buttons pushed, one ascends from k^{th} to $k+1^{st}$ floor at most one more time than one descends from the $k+1^{st}$ to k^{th} floor.

(analogous for negative floors $-k$ & $-(k+1)$)

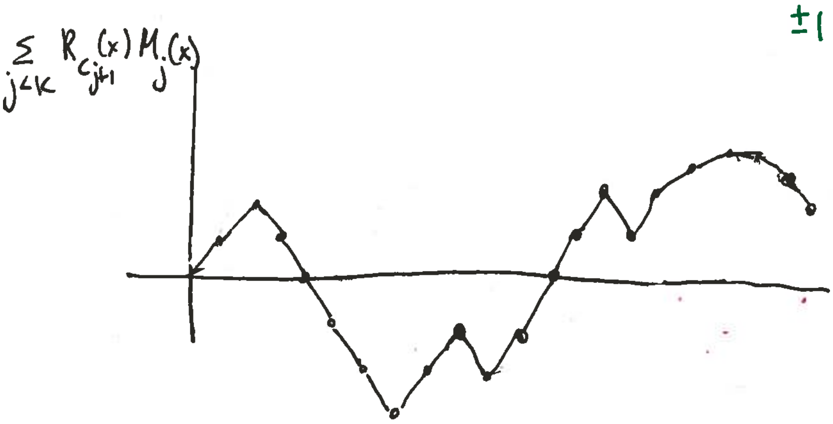
Proof by picture:



match transitions going up with those going down on same level (as in parentheses) but what is behavior of $\sum_{j<K} \underbrace{R_{c_{j+1}}}_{\pm 1}(x)\, \underbrace{M_j(x)}_{\in[0,1]}$?

$\Rightarrow |\text{slope}| \le 1$ (in fact $\le 2\gamma\varepsilon$)

+ same sign as $N_j(x)$



$$\boxed{\text{Recall: } A_i = \sum_{0\le j\le i-1} R_{c_{j+1}}(x)\, M_j(x)}$$

Matching:

For $k \ge 0$:

match $a = j$ s.t. $N_j(x) = k$ + $N_{j+1}(x) = k+1$

with $b = j'$ s.t. $N_{j'}(x) = k+1$ + $N_{j'+1}(x) = k$

For $k<0$: analogously match $-k$ to $-(k+1)$ with $-(k+1)$ to $-k$

use N_i's to create matching. the corresponding R_{c_i}'s will be such that one is $+1$ and one is -1

For each matched pair:

Will bound contribution from matched pairs by $\varepsilon\gamma$ per pair using bound on slope (and total of $\frac{\varepsilon\gamma i}{2}$)

(for each matched pair (a,b) cont.)

just by assumption that $R_{c_{a+1}}(x) = +1$ & $R_{c_{b+1}}(x) = -$

$$\underbrace{R_{c_{a+1}}(x)}_{\substack{+1 \\ \text{elevator} \\ \text{going up}}} \underbrace{M_a(x)}_{N_a(x)=k} + \underbrace{R_{c_{b+1}}(x)}_{\substack{-1 \\ \text{elevator} \\ \text{going} \\ \text{down}}} \underbrace{M_b(x)}_{N_b(x)=k+1} = M_a(x) - M_b(x)$$

if $0 \le k \le \frac{1}{\varepsilon\gamma}$ or $0 \le k+1 \le \frac{1}{\varepsilon\gamma}$

then $\underbrace{M_a(x)} - \underbrace{M_b(x)}$

$$= (1 - \varepsilon\gamma N_a(x)) - (1 - \varepsilon\gamma N_b(x))$$

$$= \cancel{1} - \varepsilon\cancel{\gamma}k - \cancel{1} + \varepsilon\gamma(\cancel{k}+1)$$

$$= \varepsilon\gamma$$

Contribution of matched edges in the "low slope" area

else $M_a(x) - M_b(x) = \begin{cases} 1-1 \\ \text{or} \\ 0-0 \end{cases} = 0$

contribution of matched edges in the "$|\text{slope}| = 1$" area

$\therefore$ each pair contributes $\le \varepsilon\gamma$ to sum

$\le \frac{i}{2}$ pairs

$\le \frac{i}{2} \cdot \varepsilon\gamma$ total contribution

Contribution from unmatched edges :

R_{c_j}

either all unmatched N_i's have negative steps

or all have positive steps

if all negative:

R_{c_j}'s all -1

M_j's all $\in [0,1]$

$\therefore$ contribution of $R_{c_{j+1}}(x) M_j(x) < 0$

if all positive :

if unmatched N_i's in $[0, \frac{1}{\varepsilon\gamma}]$

- for each $M_j \in [0,1]$, contribution of $R_{c_{j+1}} M_j(x) \le 1$

- at most $1/\varepsilon\gamma$ of these

$\Rightarrow$ total contribution $\le \frac{1}{\varepsilon\gamma}$

if unmatched N_i in $[\frac{1}{\varepsilon\gamma}, \dots]$

then $M_j = 0$

$\Rightarrow$ total contribution $= 0$

$\therefore$ Grand total $\le \frac{1}{2} \cdot \gamma\varepsilon \cdot i + \frac{1}{\varepsilon\gamma}$ $\blacksquare$