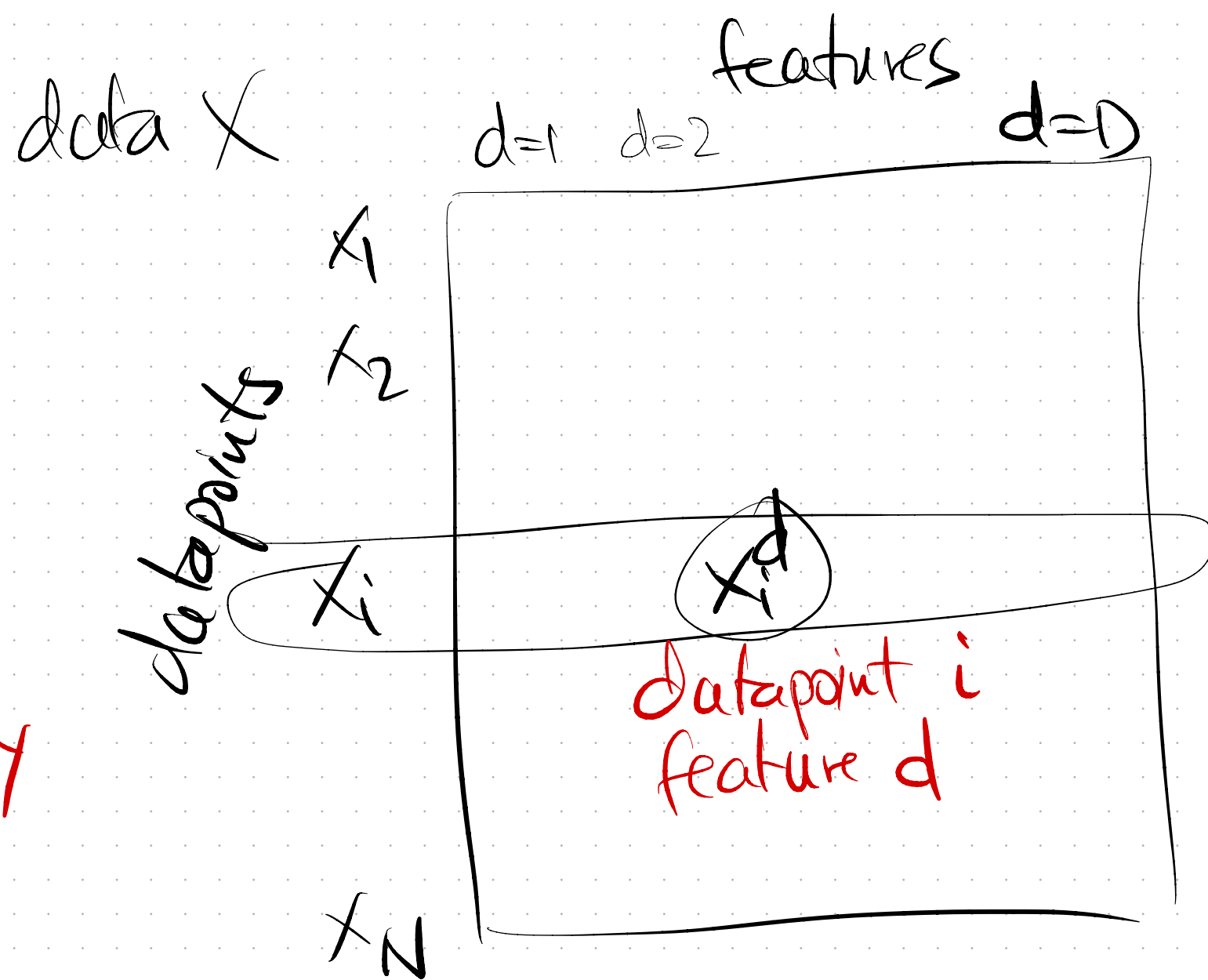


Lecture 5/21-23

- Gradient Descent
- Linear Regression with GD
- Logistic Regression
- HW 2A (released later) due 6/3 today
- ROC curve $\approx$ AUC geometrically
- Perceptron w/GD or geometry



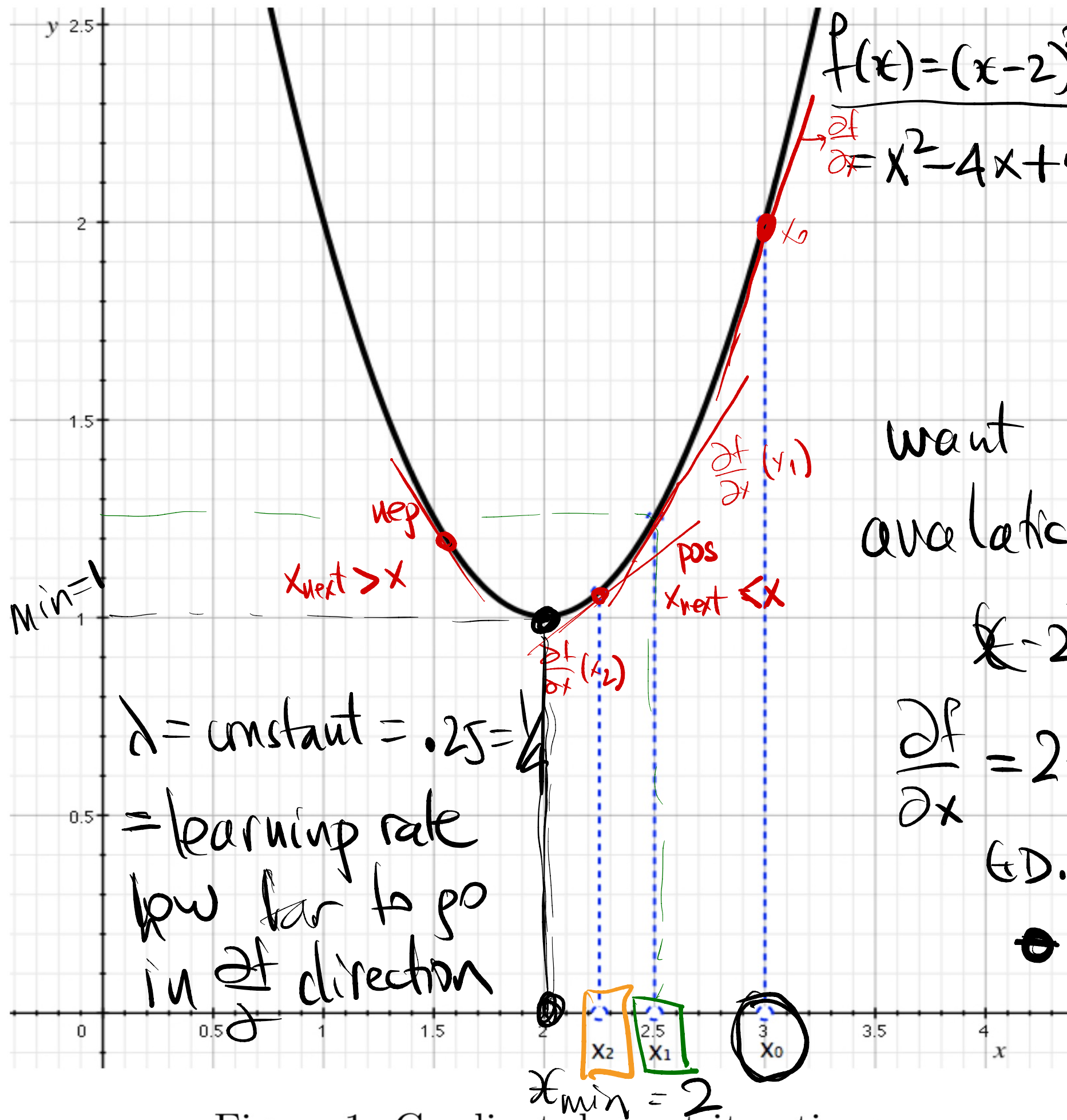


Figure 1: Gradient descent iterations

GD: iteratively find
min (or max) of $f(x)$
by moving x in direction
of gradient = differential.

want min $f(x)$ $\text{argmin} = x_{\min}$
analytic solution $x_{\min} = 2$

$$(x-2)^2 + 1 \geq 1 \text{ equality if } x = x_{\min} = 2$$

$$\frac{\partial f}{\partial x} = 2x - 4 \text{ gradient. Start at } x_0 = 3$$

GD iterations:

$$x_1 = x_0 - \lambda \frac{\partial f}{\partial x}(x_0)$$

$$= 3 - \frac{1}{4}(2 \cdot 3 - 4) = 2.5$$

- $x_2 = x_1 - \lambda \frac{\partial f}{\partial x}(x_1) = 2.5 - \frac{1}{4}(3 \times 2.5 - 4) = \boxed{2.25}$

- i want $(x_1, x_2, x_3, \dots) \rightarrow x_{min} = 2$

- What if I jump too far on the other side? $x_m < 2$

$$\frac{\partial f}{\partial x}(x_m) = 2 \cdot x_m - 4 < 0 \text{ negative}$$

update $x_{m+1} = x_m - \frac{\partial f}{\partial x}(x_m)$ increasing $x_{m+1} > x_m$

its moving x_{next} in right direction

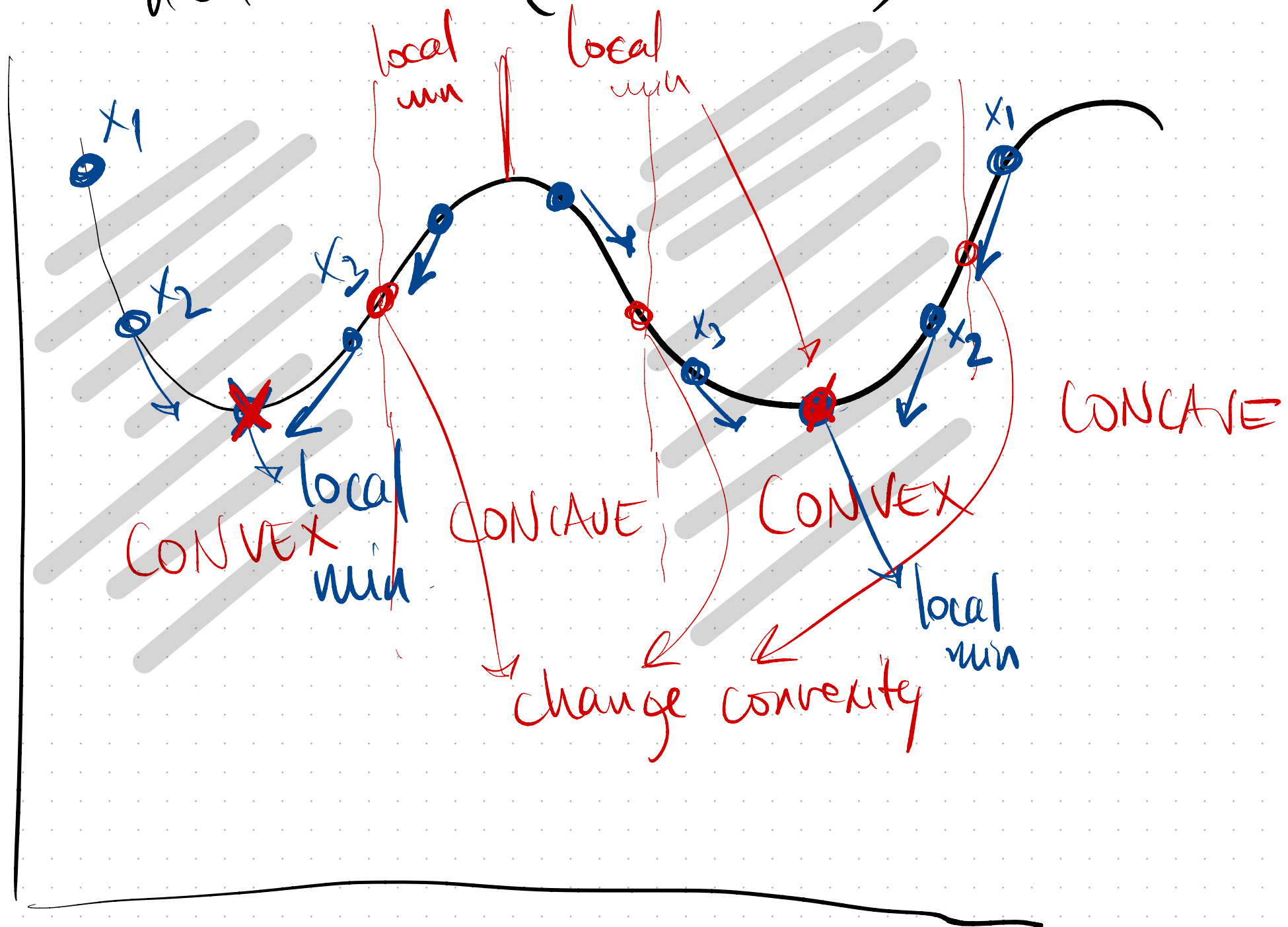
- what if $x_m = x_{min}$?

$$\boxed{\frac{\partial f}{\partial x}(x_{min}) = 0} \Rightarrow \text{update } x_{m+1} = x_m - \boxed{\frac{\partial f}{\partial x}(x_m)}$$

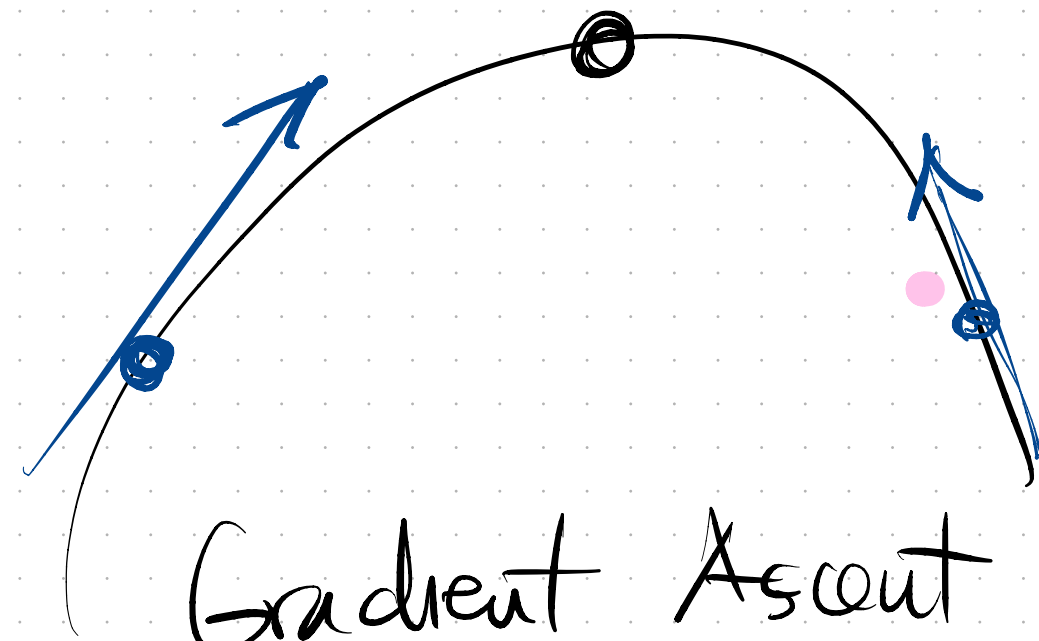
= save x_m (no change)

GD stays here

Non convex (all the time) **BUMPY**



CONCAVE Region



Gradient Ascent

$$x_{\text{new}} = x + \lambda \frac{\partial f}{\partial x}(x)$$

λ = trial and error.

- too low : many steps to converge
- too high : overshoot (other side) DIVERGE

Regression w/ GD . Error = OBJ = $J(w) = \frac{1}{2} \sum_{i=1}^N (x_i w - y_i)^2$

$$= \frac{1}{2} \sum_{i=1}^N \left(\sum_{d=1}^D x_i^d w^d - y_i \right)^2$$

$\frac{\partial J}{\partial w^d} = \frac{1}{2} \cdot 2(xw - y) \frac{\partial (xw - y)}{\partial w^d} = (xw - y) \frac{\partial (\sum_d x^d w^d - y)}{\partial w^d} = (xw - y) x^d$

w
agrup
d

GD update Rule $w_{new}^d = w^d - \lambda [(xw - y) x^d]$

(iterative)

for all d and

one datapoint x_i

$w_{new} = w - \lambda [(x_i w - y) x_i]$

vector *vector*

• batch: keep w = fixed (from prev iteration)

update $w_{new} = w - \lambda [(x_i w - y) x_i]$ for all $i = 1:N$

more
stable
batch
size $\propto N$

$\Rightarrow w_{new} = w_{(prev)} - \lambda \sum_{i=1}^N [(x_i w - y_i) x_i]$

sum of GD
updates for all
datapoints

• Stochastic : change w (current) with every datapoint update

faster convergence
"batch of size 1"
for $i = 1 : N$
 $w_{\text{new}} = w - \lambda [(x_i w - y_i) x_i]$
 $w = w_{\text{new}} \rightarrow$ before moving to next datapoint x_{i+1}

• practice: batch of size $T = 20$
batch keep w_{prev} fixed for T datapoints
update $w = \dots$

move to next batch with updated w

TERMINATION

- $\frac{\partial J}{\partial w}$ close to zero
- $|w_{\text{new}} - w|$ very low

• $J \rightarrow B$ good enough (not for regression) $\Rightarrow$ performance (acc) good enough

Reg + L_2 regularization $J(w) = \underbrace{\frac{1}{2} \sum_{i=1}^N (x_i w - y_i)^2}_{\text{data fit}} + \underbrace{\frac{\lambda}{2} L_2 \|w\|^2}_{\text{penalty}} \quad \text{Ridge}$

$$\frac{\partial J}{\partial w^d} = \dots (x w - y) x^d + \lambda w^d$$

L_2 penalty.
 $\lambda = \text{how much penalty.}$

GD update for datap x_i

$$w_{\text{new}}^d = w^d - \lambda [(x_i w - y_i) x_i^d + \lambda w^d]$$

- high L_2 penalty $\Rightarrow \|w\|$ small (constrained)
 $J(w)$ not nec. optimal
- low L_2 penalty $\Rightarrow J(w)$ minimize (optimal close to)
 $\|w\|$ large possibly.

L_1 -regularization $J(w) = \frac{1}{2} \sum_{i=1}^N (x_i w - y_i)^2 + L_1 \cdot |w|$ "Lasso"

$L_1 = \sum_{d=1}^D |w^d|$

both $L_1 + L_2$: $J(w) = \frac{1}{2} \sum_{i=1}^N (x_i w - y_i)^2 + L_1 |w| + \frac{1}{2} L_2 \|w\|^2$ "Elastic Net"

derivation L_1 -reg

$\frac{\partial J}{\partial w} = ?$ difficult

spare w
some $w^d = 0$

spread w over features

but for L_1 -reg scikit-learn
call regression

Binary Accuracy $\Rightarrow$ Confusion Matrix (2 classes yes/no)

Measure performance $\Rightarrow$ 4 types of datapoints (counts)

		TRUE LABEL	
		yes	no
PREDICTION	no	FN	TN
	yes	TP	FP
		Pos	Neg

P = pos

True Pos: pred = yes, label = yes
False Neg: pred = no, label = yes

True Neg: pred = no, label = no
False Pos: pred = yes, label = no

mistakes

Often we have score $\Rightarrow$ prediction $\hat{Y}$? use threshold θ

Score $\geq \theta \Rightarrow$ predict yes
Score $< \theta \Rightarrow$ predict no

Rank dataset $\mathcal{Z}_{\text{test}}$ by score.

TEST SET ($\mathcal{Z}$)

r_1 $h(z_1)$ = highest score
 r_2 $h(z_2)$ = second highest
 $h(z_3)$ = 3rd highest
 $h(z_k)$
 $\vdots$
 $h(z_t)$
 θ $h(z_{t+1})$
 $\vdots$
 r_T $h(z_T)$ = lowest score

predict positive

$Y(z_i) = \text{yes} = \text{pos} \Rightarrow \text{TP}$
 $Y(z_i) = \text{no} = \text{neg} \Rightarrow \text{FP}$

$$\text{ACC} = \frac{\text{TP} + \text{TN}}{|\text{test set}|}$$

predict negative

$Y(z_t) = \text{neg} \Rightarrow \text{TN}$
 $Y(z_t) = \text{pos} \Rightarrow \text{FN}$

ACC can be very sensitive to threshold θ !! $\Rightarrow$ Fix
want: measure of score performance without θ

: score to rank positives higher than negatives.

: score (positives) > scores (negatives)

ANSWER: ROC curve = curve for all thresholds θ
 $|\text{test set}| = Z = TP_{\theta} + FP_{\theta} + TN_{\theta} + FN_{\theta}$ (any θ)

$h(z)$ highest score

TP_{θ}

FP_{θ}

For thresh θ

$TPR = \text{true pos rate} = \text{recall}$

$$= \frac{TP}{\text{pos}} = \frac{TP}{TP + FN}$$

$FPR = \text{false pos rate} = \text{type I err}$

"fall out"

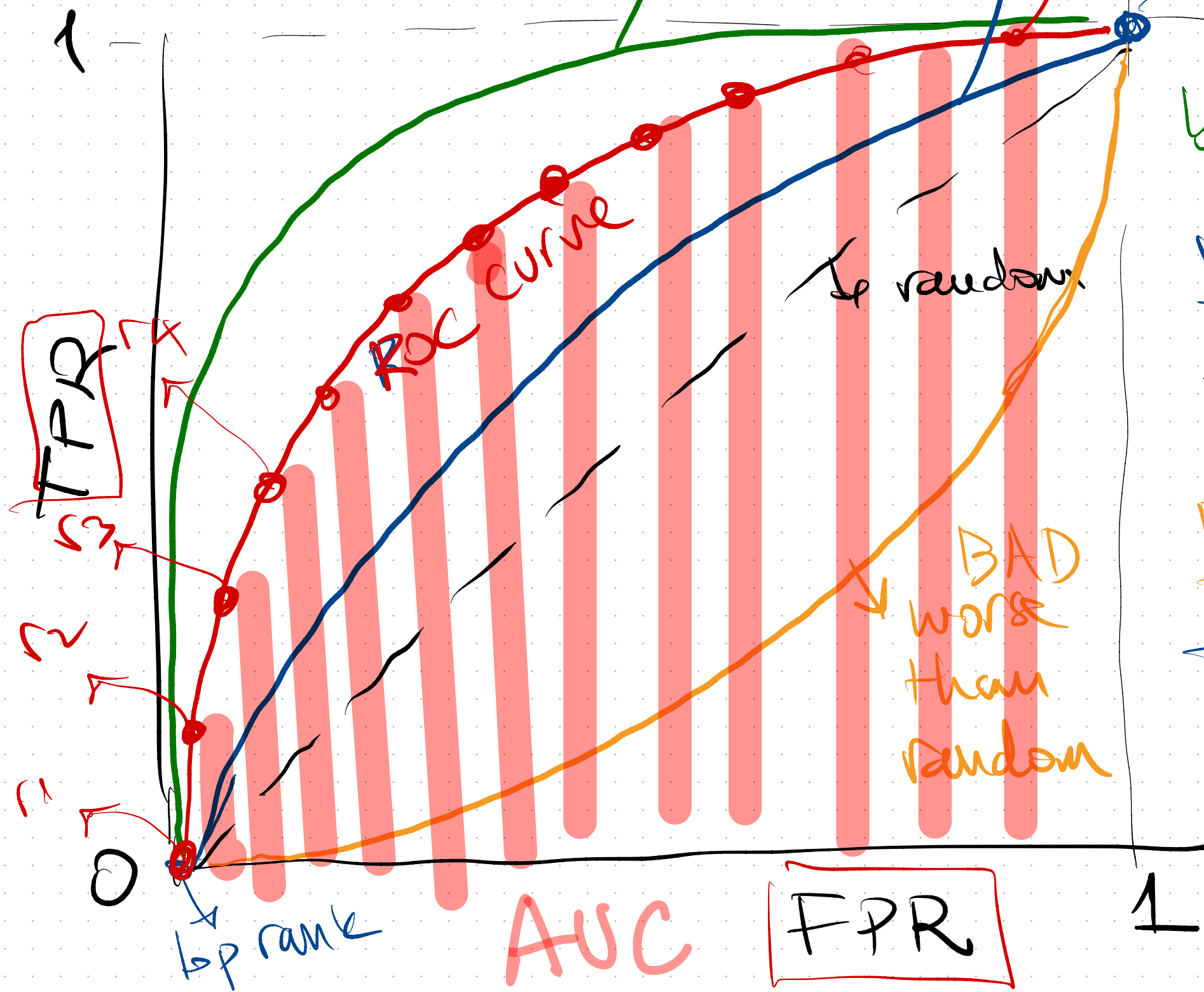
$$= \frac{FP}{\text{neg}} = \frac{FP}{FP + TN}$$

TN_{θ}

FN_{θ}

$h(z)$ lowest score

Plot TPR (y axis) vs FPR (x axis) for all thresholds Θ
 each $\Theta \approx$ a rank of 1, 2, ... $\rightarrow$ hidden on plot



better: TPR rises fast vs FPR

worse: TPR rises slow vs FPR

random predict TPR rises lin FPR

BAD: worse than random

Good performance ROC
 = green (rises fast)

MEASURE: Area under AUC Curve
 max = 1

$h(z_1)$ RANKING by score

$h(z_2) \rightarrow$ negative $y(z_2) = N$

$\vdots$

$\vdots$

$h(z_T) \rightarrow$ positive $y(z_T) = 1$

$\vdots$

} inversion (z_2, z_T)
rank incorrectly.

$$\# \text{pairs total} = \frac{T(T-1)}{2} = \# \text{correct} + \# \text{inversions}$$

inversions



$h(z_T)$

ex:

AUC

$$\frac{\# \text{correct ranked pairs}}{\# \text{total pairs}}$$

$$\text{HQA pb 5: } \text{AUC} \approx \frac{\# \text{correct pairs (in correct order)}}{\# \text{total pairs}} = \% \text{ correct ranked pairs}$$

Score-sort Ranking (highest = top) labels = $y_1, y_2, \dots, y_N$ binary $\in \{+, -\}$
 Ranks Score Label At threshold Θ for scores $\Leftrightarrow k$ for ranks

Ranks	Score	Label
$k=1$	$y_1 = \oplus$	$\rightarrow TP$
$k=2$	$y_2 = \oplus$	
$k=3$	$y_3 = \ominus$	$\rightarrow FP$
	$y_4 = \oplus$	$\rightarrow TP$
	$\vdots$	

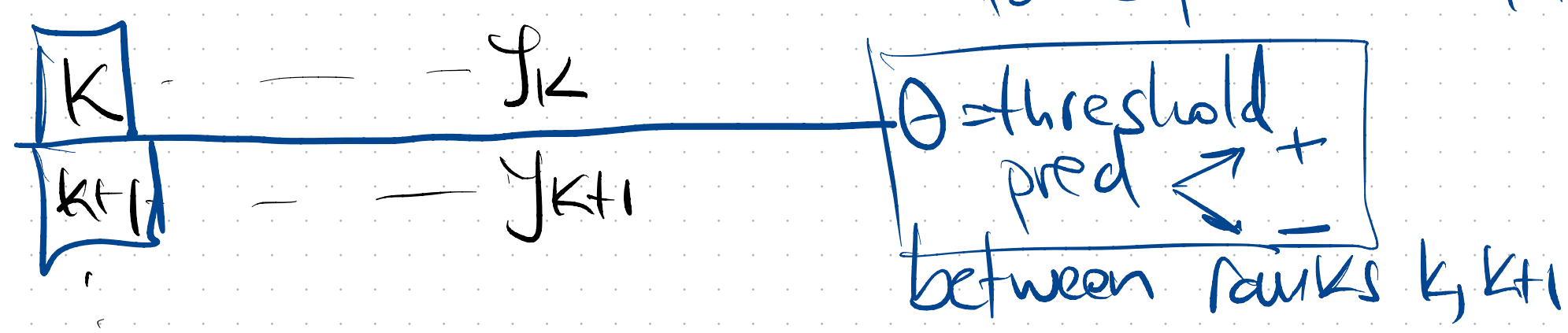
$$TPR_k \text{ true pos rate} = \frac{TP(k)}{\text{all pos}} =$$

$$FPR_k \text{ false pos rate} = \frac{FP_k}{\text{total neg}} =$$

$$\sum_{i=k}^{top} [y_i = +] \quad \begin{matrix} 1 \\ \text{or} \\ 0 \end{matrix}$$

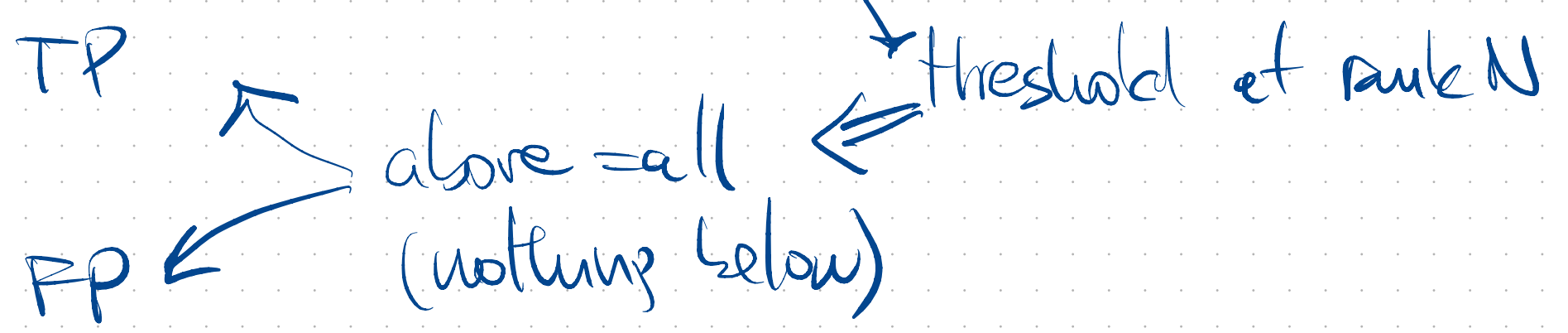
$$\sum_{i=k}^{top} [y_i = -] \quad \begin{matrix} N_+ \rightarrow \text{total pos} \\ N_- \rightarrow \text{total neg} \end{matrix}$$

$\# \text{ labels negative above threshold}$

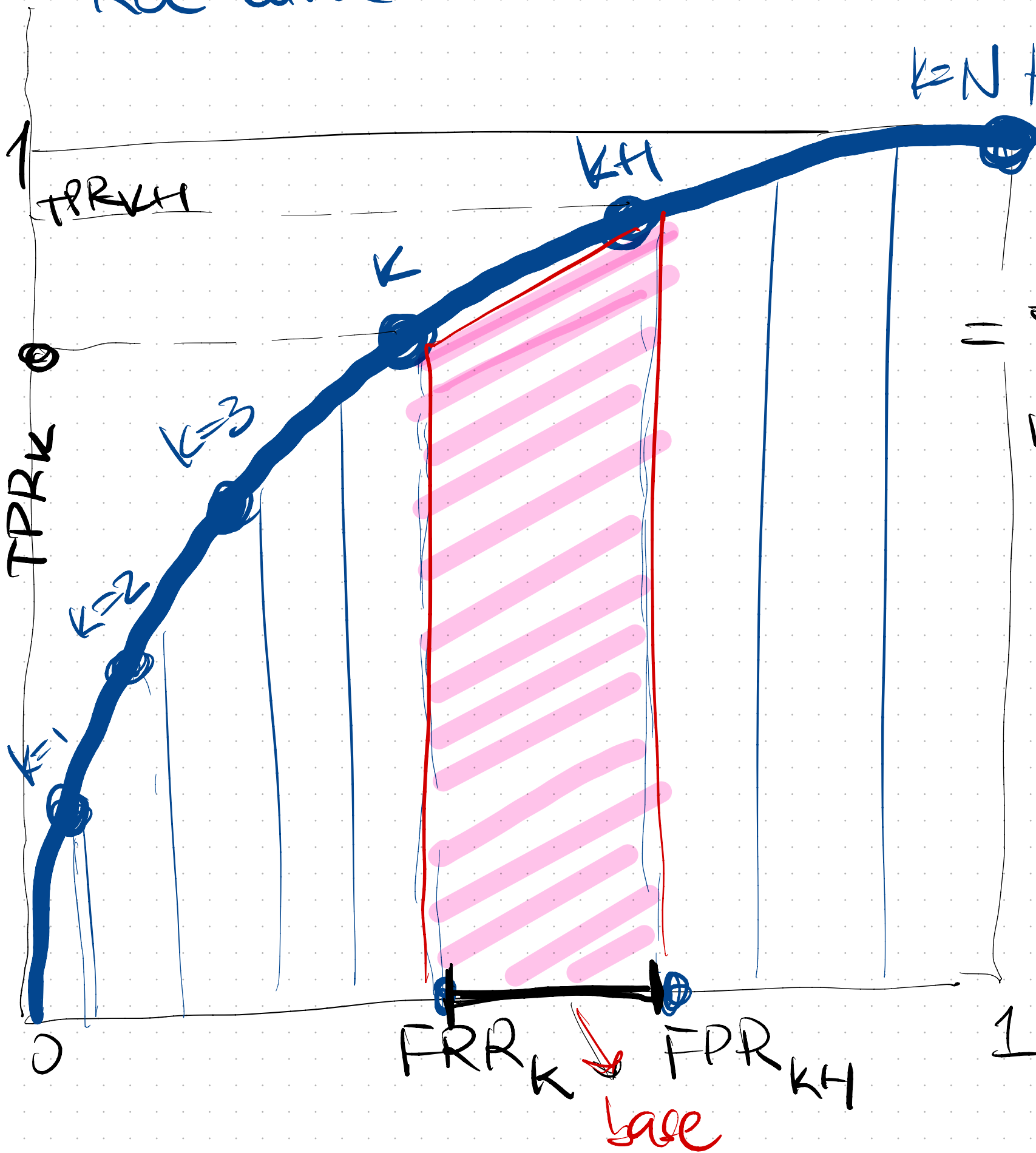


$N-1$	$y_{N-1} = \oplus \rightarrow \text{false neg FN}$
N	$y_N = \ominus \rightarrow \text{true neg TN}$

$$\begin{aligned} N_+ &= TP + FN = \text{total pos} \\ N_- &= TN + FP = \text{total neg} \\ N_+ + N_- &= N \end{aligned}$$



ROC curve



$$AUC \approx \sum \text{area-trapezoids} = \sum_{k=1}^N \left(\frac{TPR_k + TPR_{k+1}}{2} \right) \cdot (FPR_{k+1} - FPR_k)$$

height

base

$$= \sum_{k=1}^N \frac{[y_i = +] + \frac{1}{2}[y_{k+1} = +]}{N_+} \cdot \frac{[y_{k+1} = -]}{N_-}$$

want "prop of correct rank pairs"

top

$\sum_{i=k+1}^N [y_i = +]$

ranked above k pos

$\sum_{k=1}^N \frac{N_+}{N_-}$

all pairs (+ -)

every neg