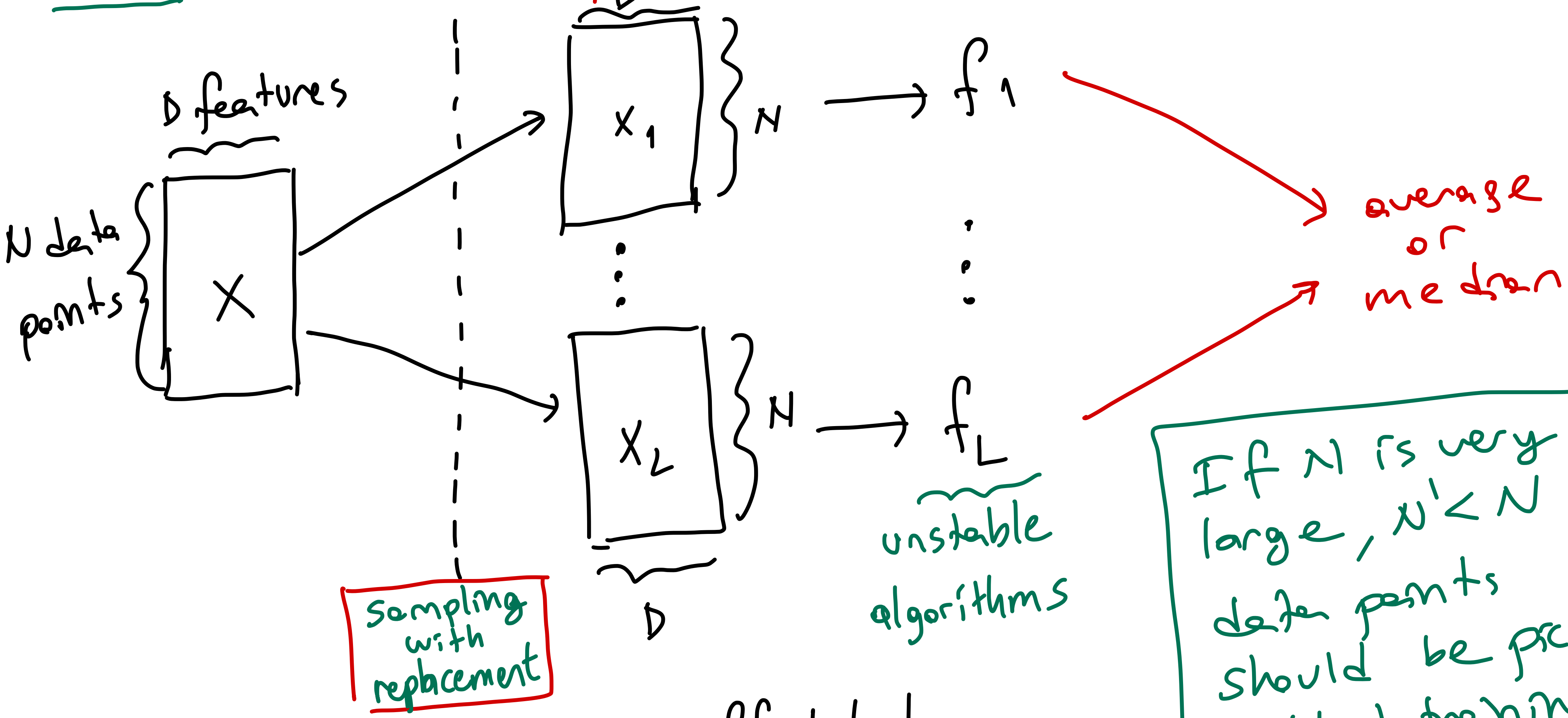
BAGGING (Bootstrap AGGregation)



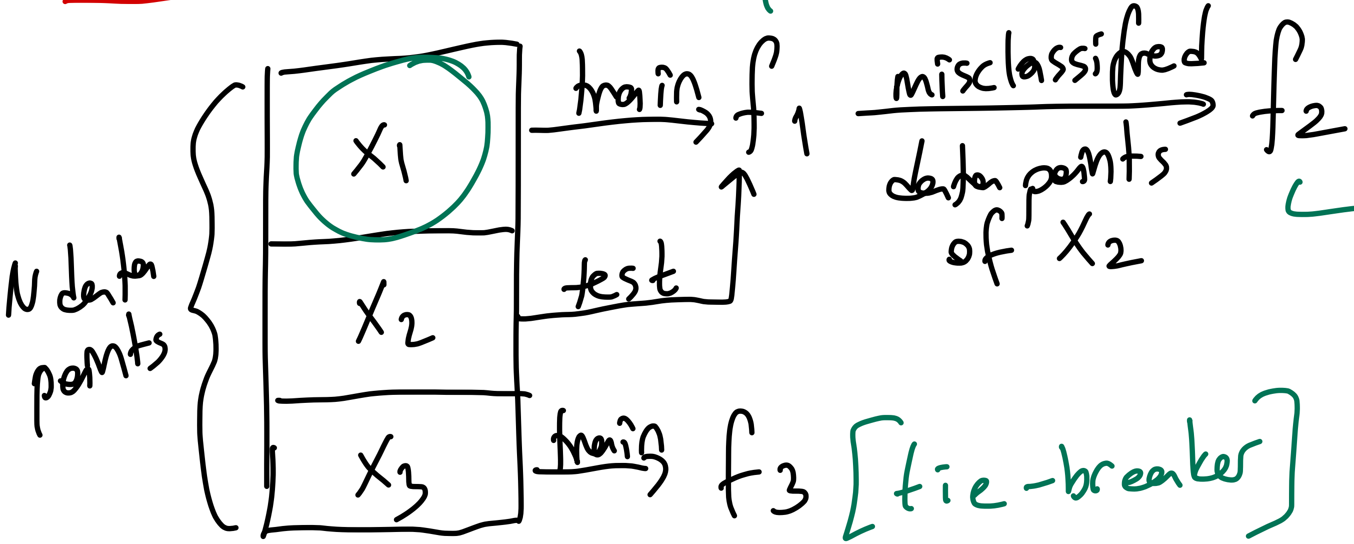
If N is very large, N' < N data points should be picked so that training sets would become different enough.

Unstable Algorithm: Highly affected by small changes in the training data set.

Unstable ⇒ DT

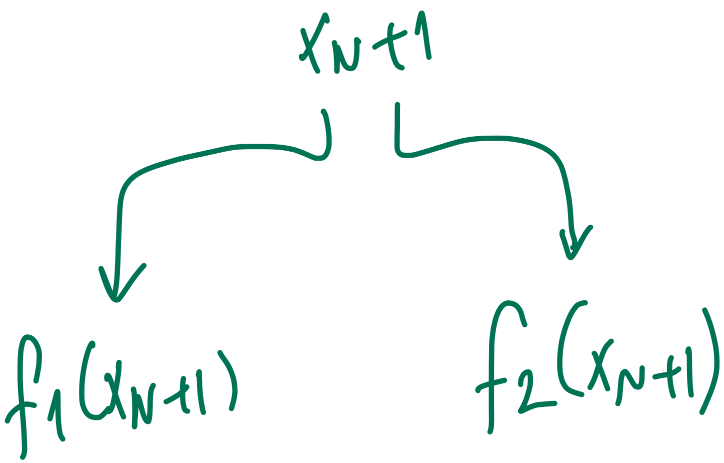
Stable ⇒ k-NN

BOOSTING:



↳ it uses the whole X_1 during training

↳ it uses a subset of X_2 during training.



$f_1(x_{N+1})$ and $f_2(x_{N+1})$

① If they agree on their decisions, no problem and use the predicted class label.

② If they donot agree, use $f_3(x_{N+1})$ as the predicted class label.

AdaBoost: modify the probabilities of drawing instances as a function of the error.

P_{ij} = the probability that the data point x_i is selected (used in training) by classifier f_j

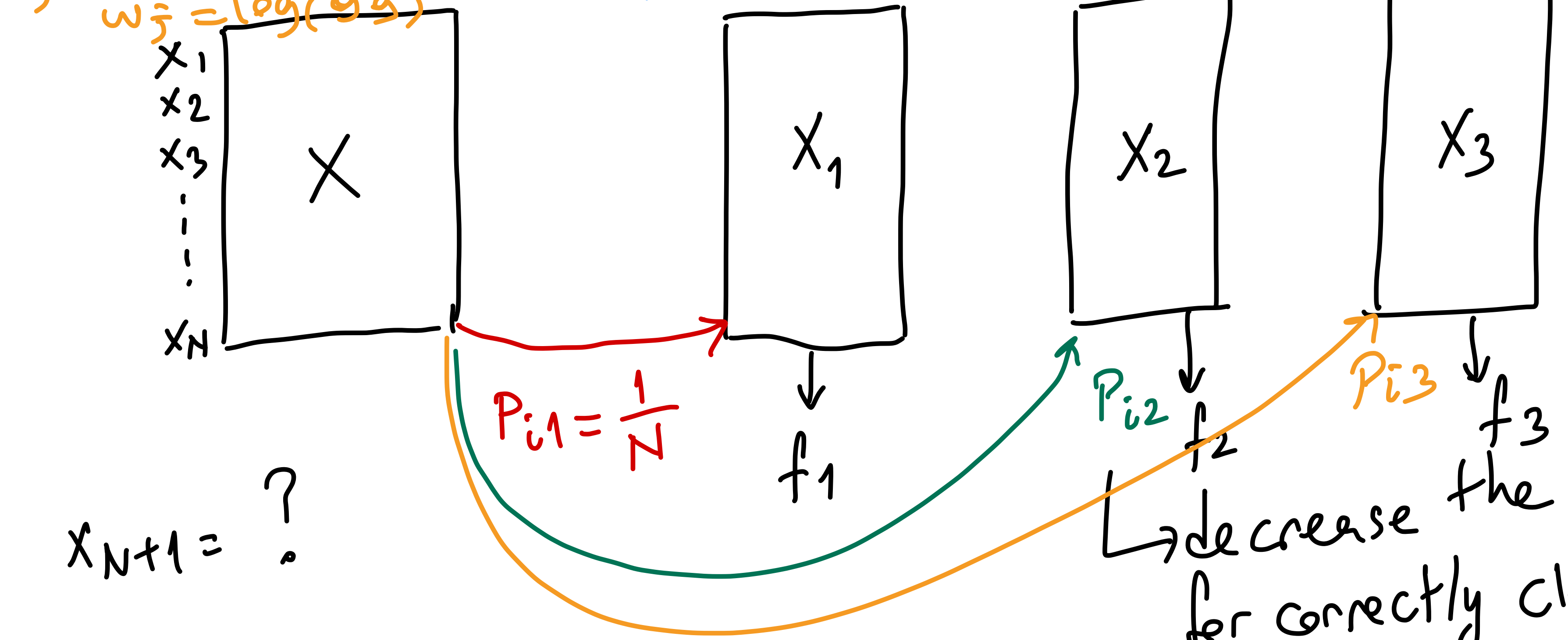
error rate $\rightarrow$ $\epsilon_j = 0.2$

$$\beta_j = \frac{\epsilon_j}{1-\epsilon_j} = \frac{0.2}{0.8}$$

$$w_j = \log\left[1/\beta_j\right] = \log(4)$$

$\epsilon_j = 0.01$ $\beta_j = 1/99$ $w_j = \log(99)$

$\epsilon_j = 0.5$ $\beta_j = 1$ $w_j = \log(1) = 0$



$x_{N+1} = ?$

↳ decrease the probabilities for correctly classified data points

↳ increase the probabilities for incorrectly classified data points

$$f(x_{N+1}) = w_1 f_1(x_{N+1}) + \dots + w_L f_L(x_{N+1})$$

based on their error rate.

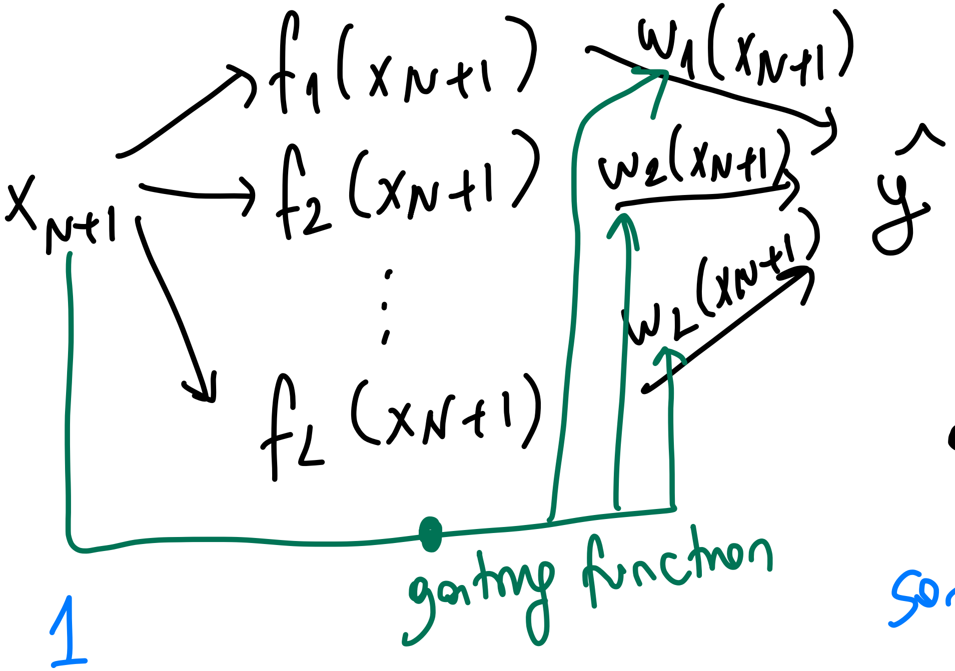
Mixture of Experts (MoE):

Voting ⇒ $\hat{y} = \sum_{j=1}^{L} \boxed{w_j} f_j(x_{N+1})$ → Constant over the input space.

MoE ⇒ $\hat{y} = \sum_{j=1}^{L} \underbrace{w_j(x_{N+1})}_{} f_j(x_{N+1})$ → w_j's will be assigned by the gating function

Cooperative

- $w_1, w_2, \ldots, w_L$ are assumed to be independent.



Competitive

- $w_1, w_2, \ldots, w_L$ are producing sperse weights (mostly zero)
- one or some of them are nonzero.

sigmoid

$$w_j = \frac{1}{1+\exp[-(v_j^T x + v_{j0})]}$$

softmax

$$w_j = \frac{\exp(v_j^T . x + v_{j0})}{\sum_{k=1}^{L} \exp(v_k^T . x + v_{k0})}$$

Stacked Generalization

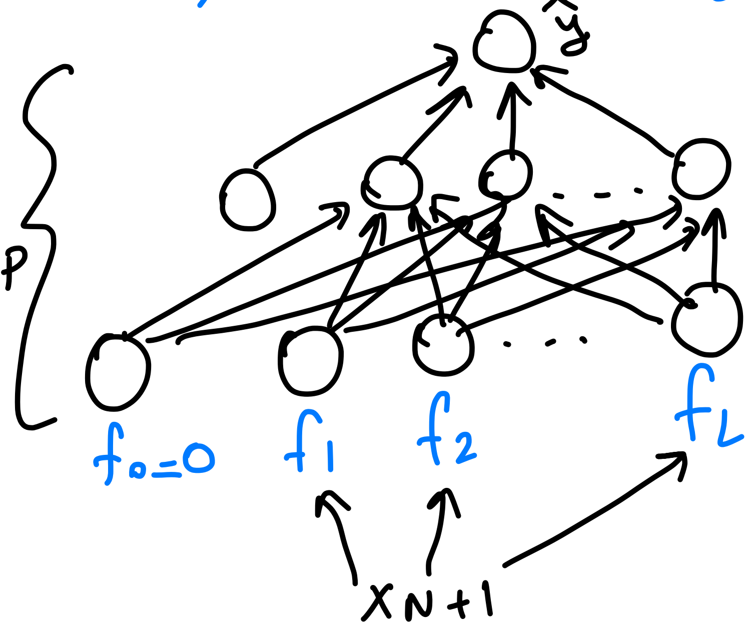
Voting ⇒ $\hat{y} = \sum_{j=1}^{L} w_j f_j(x_{N+1})$

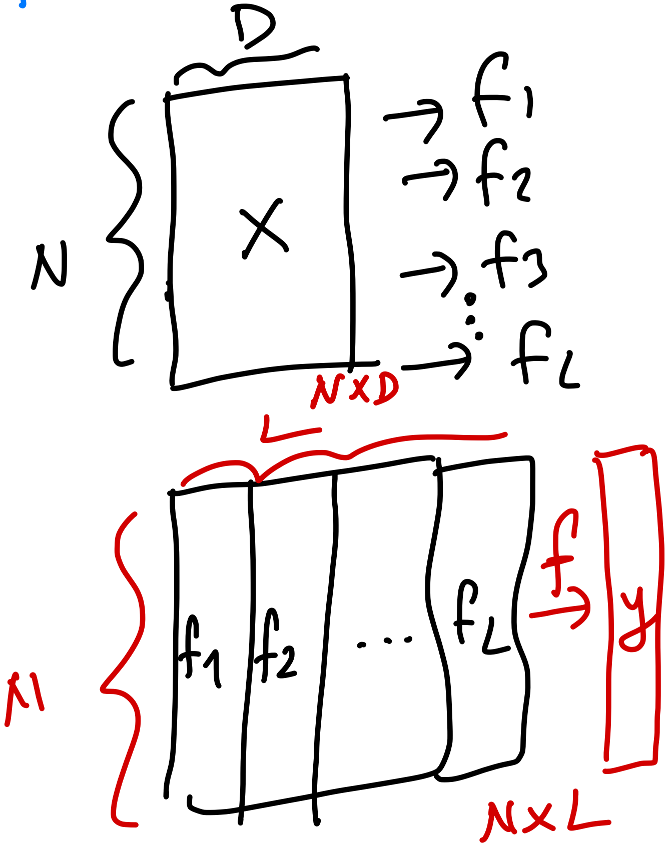
MoE ⇒ $\hat{y} = \sum_{j=1}^{L} w_j(x_{N+1}) f_j(x_{N+1})$

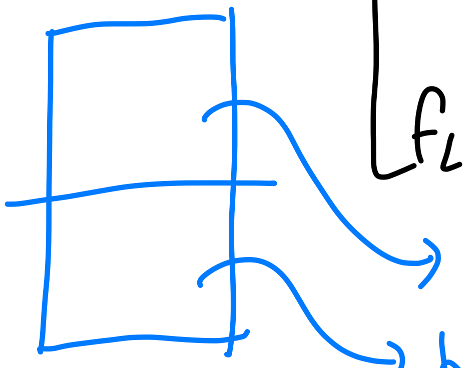
Stacked Generalization ⇒ $\hat{y} = f\left(f_1(x_{N+1}), f_2(x_{N+1}), \ldots, f_L(x_{N+1}) \right)$

↳ nonlinear algorithm

$$\underset{D\times 1}{(x_{N+1})} \Rightarrow \underset{L\times 1}{\begin{bmatrix} f_1(x_{N+1}) \\ f_2(x_{N+1}) \\ \vdots \\ f_L(x_{N+1}) \end{bmatrix}}$$



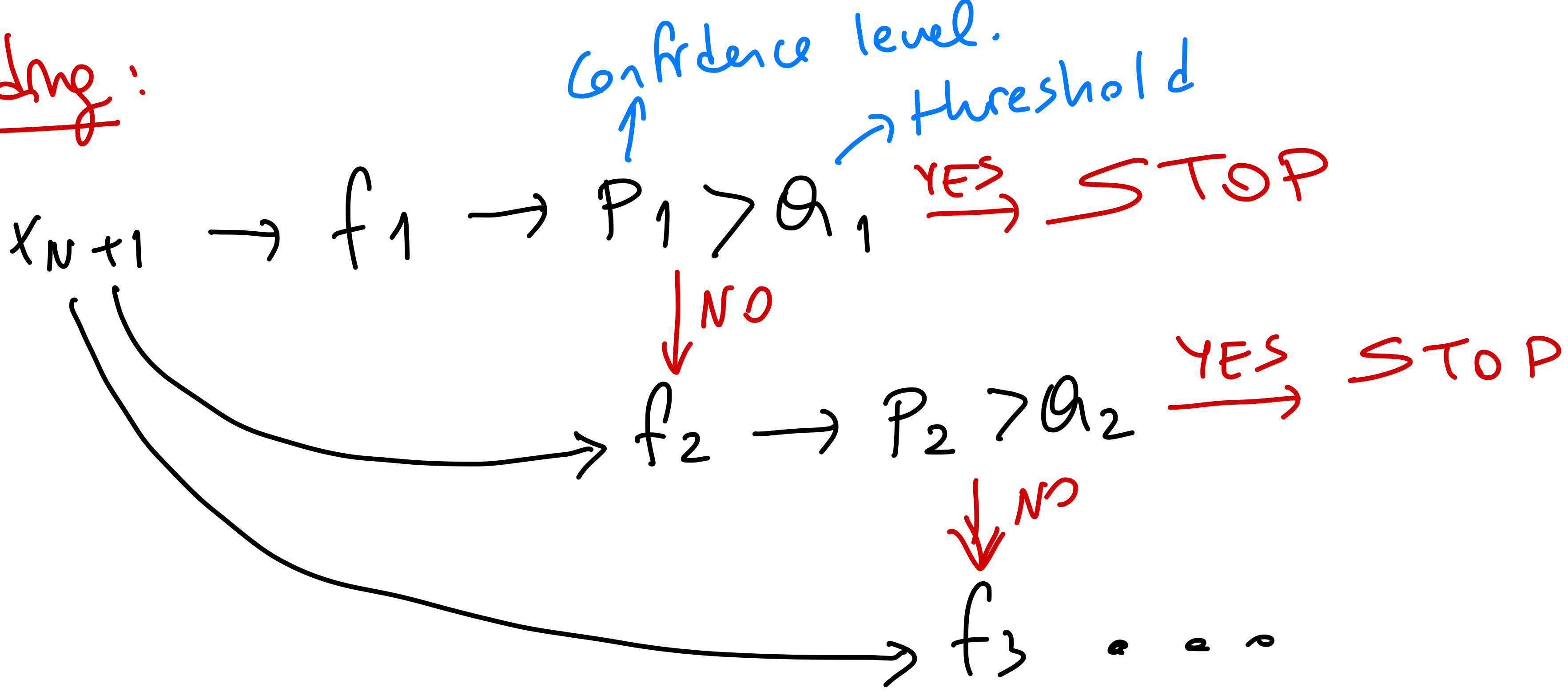




training base learners

training the combination algorithm

Cascading:



$$\overset{0.95}{\theta_1} > \overset{0.90}{\theta_2} > \overset{0.85}{\theta_3} > \dots$$

decreasing thresholds

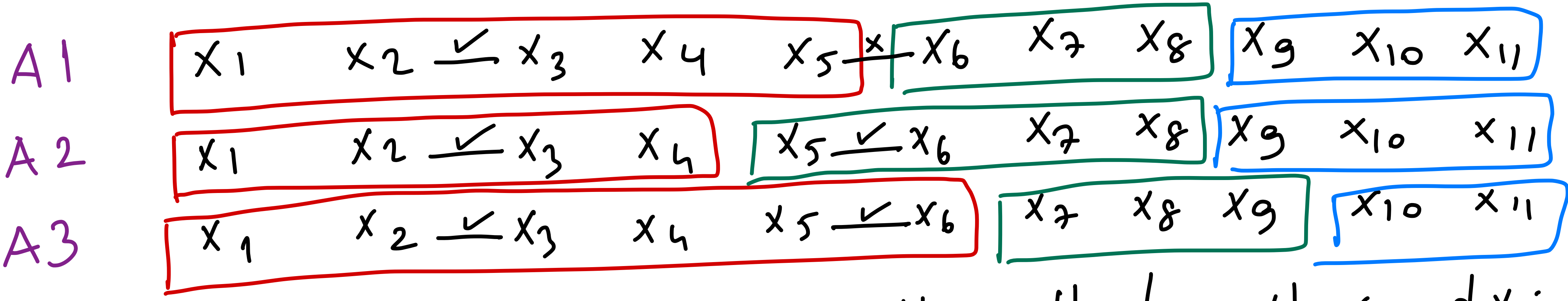
$x_{N+1} \xrightarrow{f_1} p_1 = 0.98 \Rightarrow$ we are confident enough, let's STOP.

since $0.98 > 0.95$

$x_{N+1} \xrightarrow{f_1} p_1 = 0.92 \xrightarrow{f_2} p_2 = 0.91 \Rightarrow$ we are confident enough, let's STOP.

since $0.92 < 0.95$

since $0.91 > 0.90$


A1
x1 x2 ✓ x3 x4 x5 × x6 x7 x8 x9 x10 x11
A2
x1 x2 ✓ x3 x4 x5 ✓ x6 x7 x8 x9 x10 x11
A3
x1 x2 ✓ x3 x4 x5 ✓ x6 x7 x8 x9 x10 x11

A^{*}

C_{ij} = # of clustering algorithms that put x_i and x_j into the same cluster.

$C_{5,6} = 2$

$C_{2,3} = 3$

$$C = \begin{array}{c|ccccccccccc} & x_1 & x_2 & x_3 & x_4 & x_5 & x_6 & x_7 & x_8 & x_9 & x_{10} & x_{11} \\ \hline x_1 & 3 & 3 & 3 & 3 & 2 & & & & & & \\ x_2 & 3 & 3 & 3 & 3 & 2 & & & & & & \\ x_3 & 3 & 3 & 3 & 3 & 2 & & & & & & \\ x_4 & 3 & 3 & 3 & 3 & 2 & & & & & & \\ x_5 & 2 & 2 & 2 & 2 & 3 & 2 & & & & & \\ x_6 & & & & & & & & & & & \\ x_7 & & & & & & & & & & & \\ x_8 & & & & & & & & & & & \\ x_9 & & & & & & & & & & & \\ x_{10} & & & & & & & & & & 3 & \\ x_{11} & & & & & & & & & & & 3 \end{array}$$

Clustering on the C matrix would give you A^{*}.

11x11