Week 13: (9-4 - 9-5) Combining Models:

Learning objectives:
- Advantages of combining models
- Ways to combine models
- Explain decision tree learning
- Explain adaptive boosting (adaboost)

Questions?:

1. Why to combine models?
   - Multiple available, none perfect
   - Many features extracted from given pattern
   - Complimentary properties among different models and features

2. Issues:
- How many classifiers are needed?
- What kind of model should be used?
- Feature used in classifier?
- How to combine results from different classifiers?

Stories of success:

Million dollar prize: Netflix top submissions combine models.

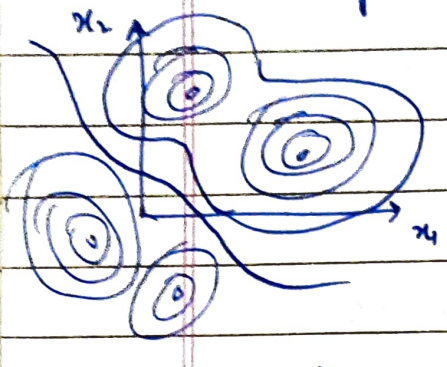
Data mining competitions: Classification problems.

Winning team employ ensemble of classifiers

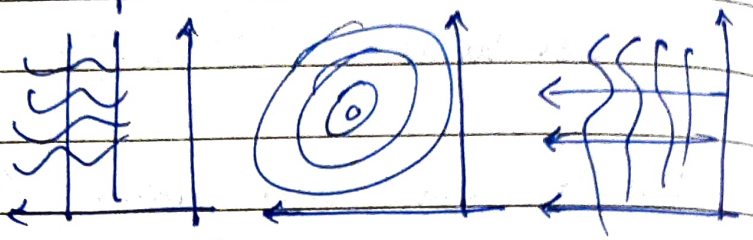
Ensemble ⇒ Averaging multiple models
↳ Gives global picture

PAGE No.
DATE / /

Ex.) Combining two 1D Gaussians

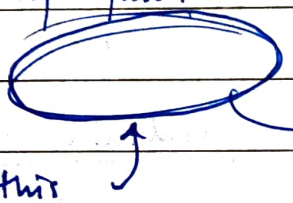


Combined decision boundary for multiple models needed ?



Introduction to hypothesis class:

Space of functions : Set of functions modelled by 1 model



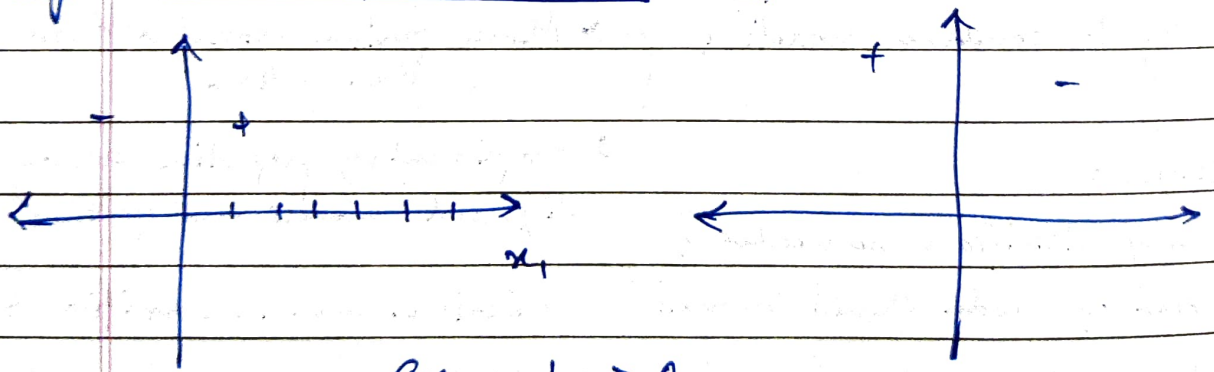
Comes close but not exactly models data

$h_\theta(x)$ richer set may be a superset of this → Hypothesis class

Eg. $h_\theta(x)$ all linear fns $g_\phi(x)$ all polynomials

$\Rightarrow g_\phi(x) \supset h_\theta(x)$

Hypothesis class of a threshold:



$ax_1 + b > 0$

Only one variable can form threshold.

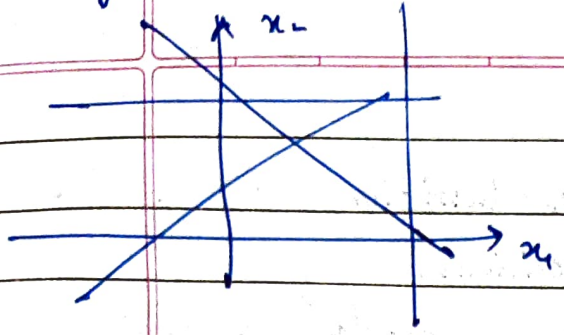
Eg. $x_1 < \text{or} > a_1$

or $x_2 < \text{or} > a_2$

Only line ∥ to one of the axis

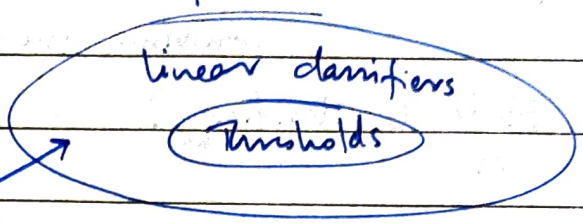
(No slant threshold boundary allowed)

Hypothesis class of a linear classifier.



$w^T x + b > 0$

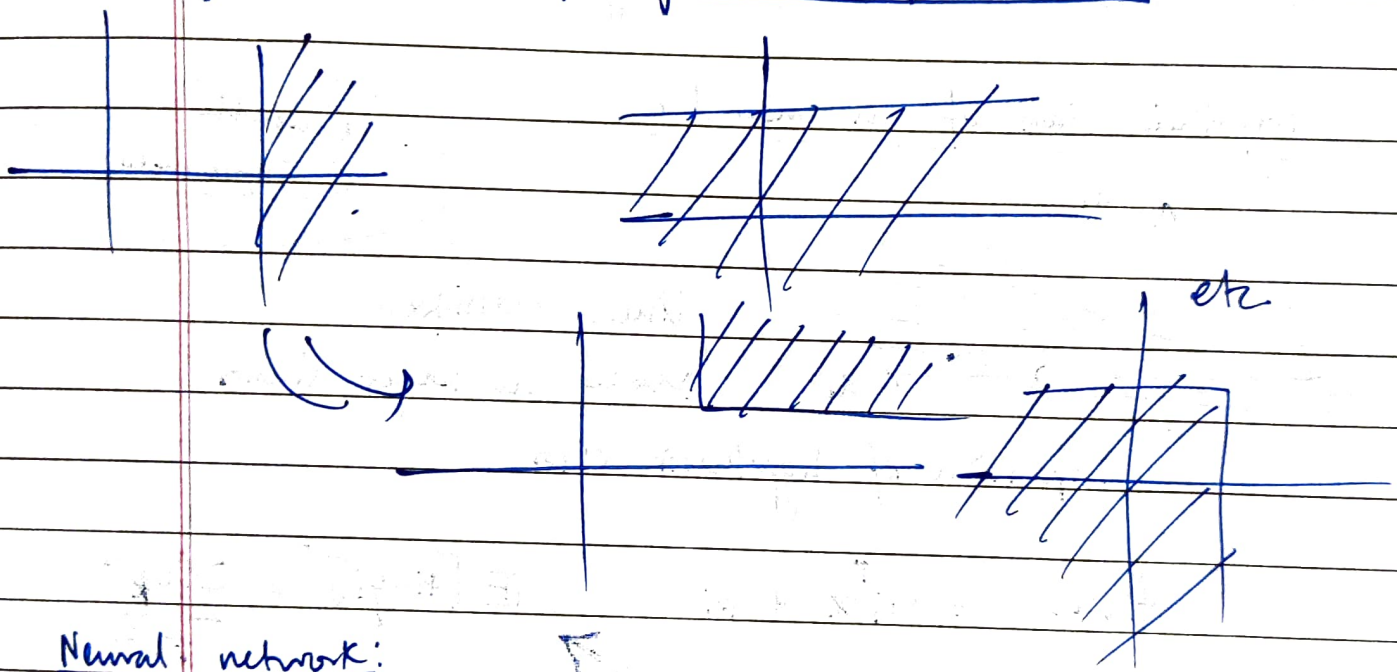
Kind of decision thresholds



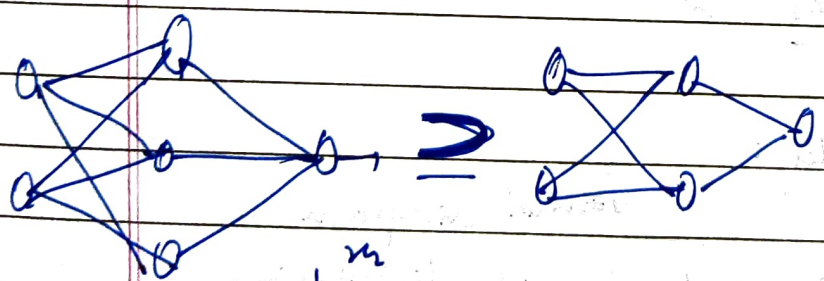
loss + $\frac{\lambda}{2}\|w\|^2$

reduces hypothesis class

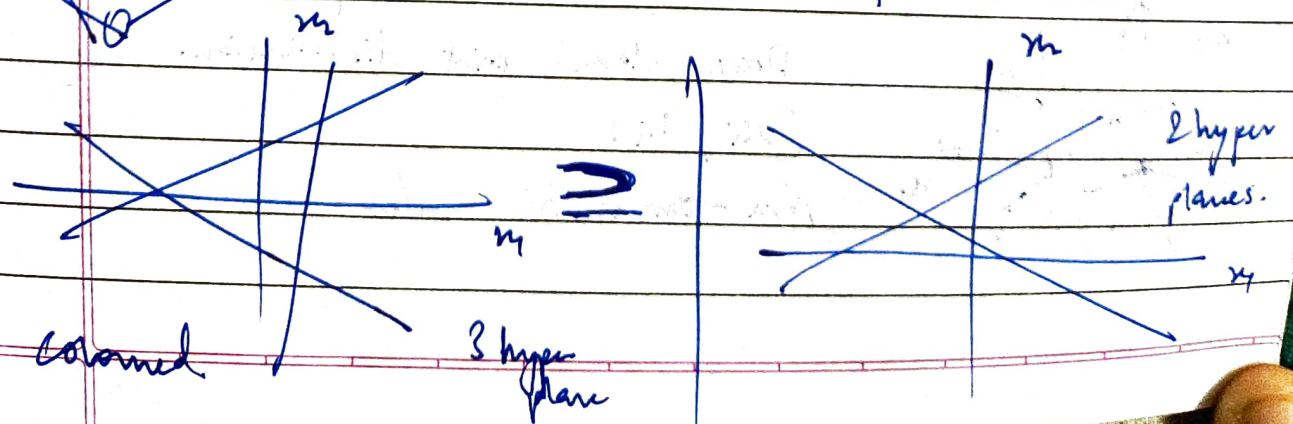
Nested hypothesis classes of single & multiple variables



Neural network:



Can get 2 hidden NN from 3 one



(9-2) Ensembles :

Ensembles for classification

Classifier 1 ← | → Classifier 2 ... n

Combine their decisions.

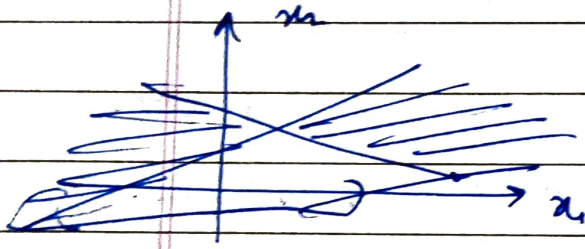
→ Mode (classifier) $y = \underset{i=1}{\overset{n}{\text{mode}}} f_i(x)$

Works if they are independent of each other. (Assumption of independence)

Binomial distribution. (Obviously)

Hypothesis class of an ensemble :

Ensemble style model



Linear classifier

Mode no longer linear

Expansion of hypothesis class

$f_i(x) = w_i^T x + b_i$

$\mathbb{E}[t_i - y_i]^2 = \frac{\sum \sigma_k^2}{K^2}$

$\therefore y_i = \frac{\sum f_i(x)}{K}$

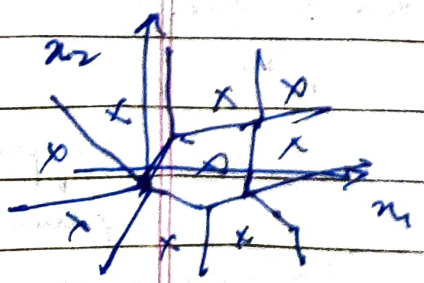
Nearest neighbor model :

- Voronoi diagram

Draw closest point distributions

(x_i, t_i)

Non-parametric model



Combine K models by:-

K-Nearest Neighbor model

PAGE No.
DATE / /

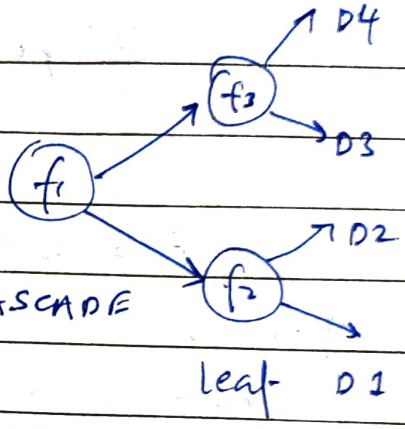
If some value is wrong, it is corrected & averaged out- Hypothesis class have a smoothening & self correcting effect

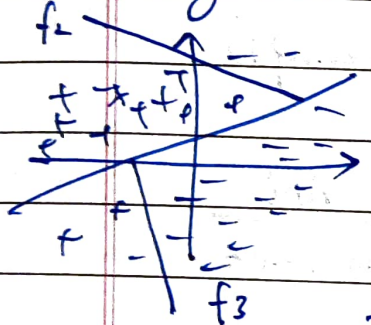
(9-3-1) Decision Trees 1

Voting

f_1 f_2 f_3 = Ensemble

In decision tree, we first use f_1 such that Each classifier may be weak but combined may be better.



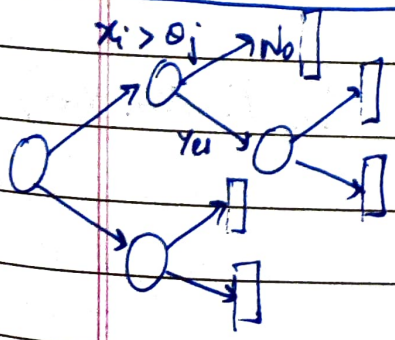


This is how cascade seperates the classes.

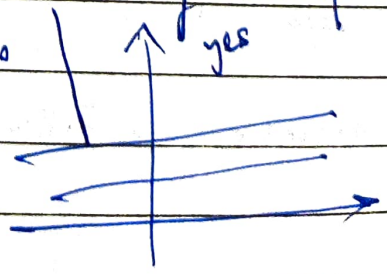
It is a tree as last decision nodes = leaf & Earlier levels are internal nodes.

— Need not be balanced, need not be binary either

Threshold-based decision tree:



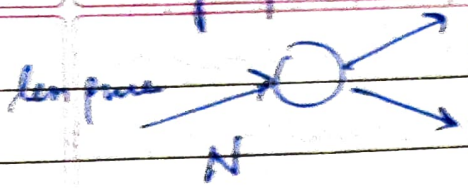
Binary decision at every node of tree



Greedy algorithm for node splitting

PAGE No.
DATE / /

- Purity of branches to be maximized.



Predominance of one class over the other classes

$\max_c (p_c)$ probability of class c $\quad c \in \{1, \ldots, C\}$

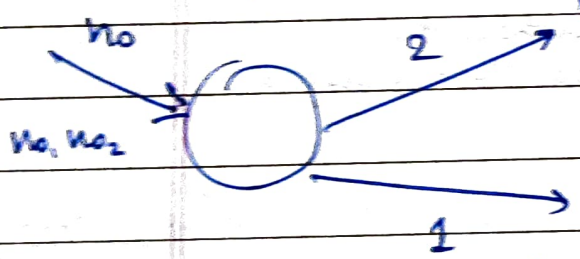
$\sum_c p_c = 1 \qquad p_c \geq 0$

$-\sum p_c \log(p_c)$ (impurity) Entropy minimized?

$p_c = n_c / n$ Or sum $\sum p_c (1 - p_c)$ Gene index

(9-3-2)

Greedy: Balanced purity (Need this concept)



$p_{0i} = \dfrac{n_{0i}}{n_{01} + n_{02}}$

$p_{21} = \dfrac{n_{21}}{n_{11} + n_{22}}$

$p_{11} = \dfrac{n_{11}}{n_{11} + n_{12}}$

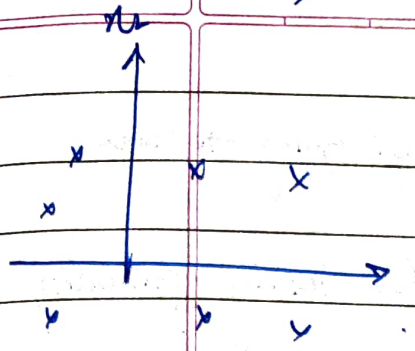
p_{11} & p_{21} need to encode MORE purity.

$\Rightarrow W_1 H(p_{11}, p_{12}) + W_2 H(p_{21}, p_{22}) < W_0 H(n_{01}, n_{02})$

node node

Max decrease $\Rightarrow$ Better decision update.

We have to look at all possible x, a relations to create thresholds



D features Gaps between N points = N-1

$\Rightarrow$ D x (N-1) decisions made at any node

$D(N-1) \rightarrow D(N_1-1) + D(N_2-1)$

Algorithm:

- Start at root
- Find ideal x, a relation to separate / max. purity
- If pure, label & terminate
- Recurse over children node.

Issue: Tree gets deeper such that nodes $\propto$ N (samples)
Too many axis aligned boundaries. $\Rightarrow$ Not a good hypothesis class

Depth of tree $\propto$ Overfitting $\Rightarrow$ Pruning needed
Validation data can be used, etc.

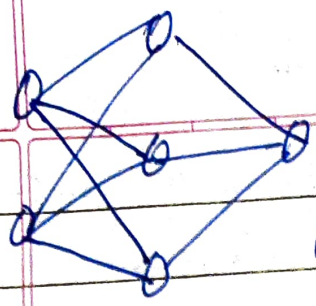
(9-4) Combination of trees. Model randomization, Random Forests

Tree 1 + Tree 2 + Tree 3 / Average / mode

$\Rightarrow$ Ensemble of trees

How to find threshold such that to have min. mean square error on children $\equiv$ Random Forest = Ensemble of trees.

Randomizing Models $\rightarrow$
- Non convex initialization
- Variable subset selection
- Training subset selection $\Rightarrow$ Bagging.



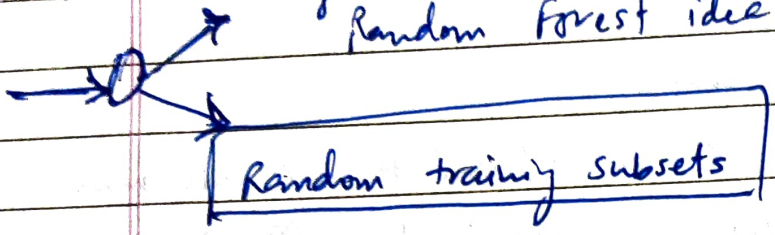
Random initialization of Neural Network

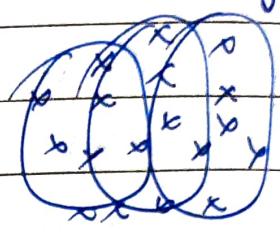
PAGE No.
DATE / /

At every node we are not going to use all D variables

⇒ Randomly select $d < D$ variables for testing optimality

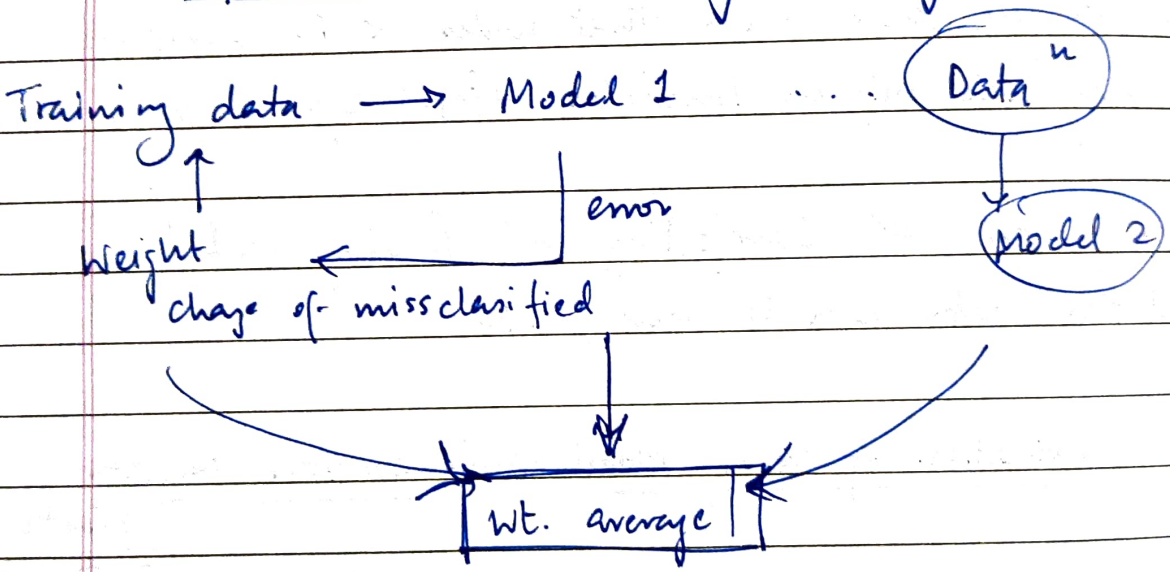
Random Forest idea





(9-5) Boosting:

Sequential refinement using boosting



Adaboost:

- Uniform weight for all data points
- At each round
  - Bootstrap sample based on the weights
  - Train classifier & apply to original train set
  - Wrongly classified ⇒ Wt increased
  - Correctly classified ⇒ Wt decreased

Error > 50% start over

- Final prediction is weighted avg of all classifiers with weights representing the training accuracy.

PAGE No. / DATE / /

Adaboost visualized:

S is training data

Initialize $D^{(1)} = \left[\frac{1}{n}, \ldots \frac{1}{n}\right]$

for t in 1 ... T / m in 1 ... M

↳ Learn t^{th} WL on $(D^{(t)}, S)$

(Weighted loss)

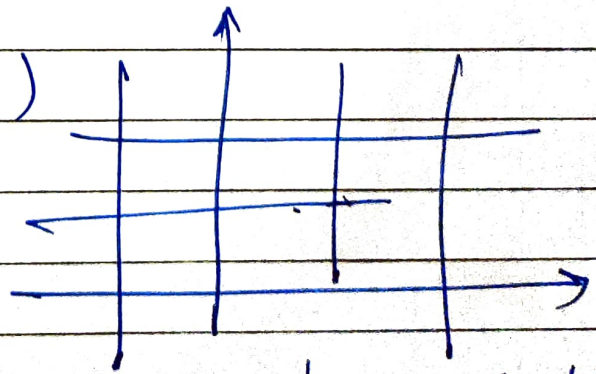
$$\varepsilon_t = \sum_i D_i^{(t)} \mathbb{1}\{y_i \neq t_i\} \qquad \alpha_t = \frac{1}{2} \log\left[1/\varepsilon_t - 1\right]$$

All training points, recompute wts st we focus on better performing on mis-ed points

$$\Rightarrow D_i^{(m+1)} = \frac{D_i^{(m)} \exp(-w_t t_i y_i)}{\sum_j D_j^{(m)} \exp(-w_t t_j y_j)}$$

Proceed for T steps.

$$\text{Sign}\left(\sum_{m=1}^{M} w_m y_m^{(t)}\right)$$



Many learners combine & give ensemble

Adaboost:

Viola Jones face detector

Using Haar filters