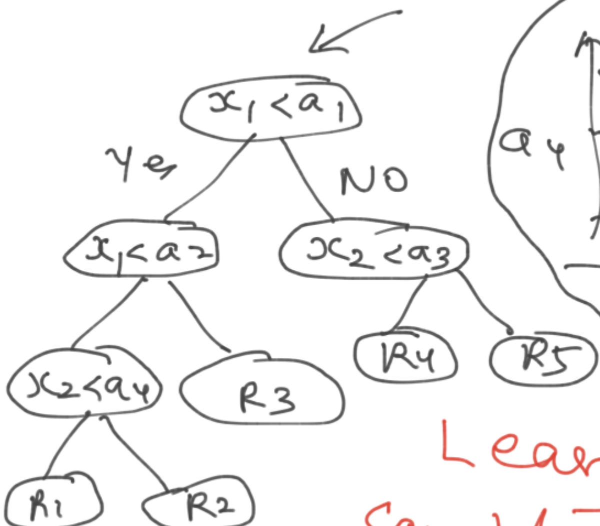
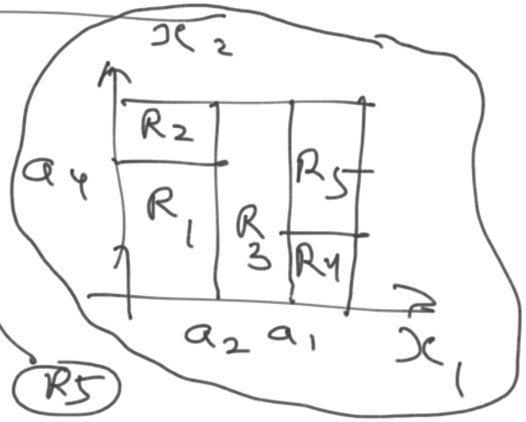
Lecture 13 FML April 6, 21

multiclass classification

Decision Trees





Learning the tree

Sample size

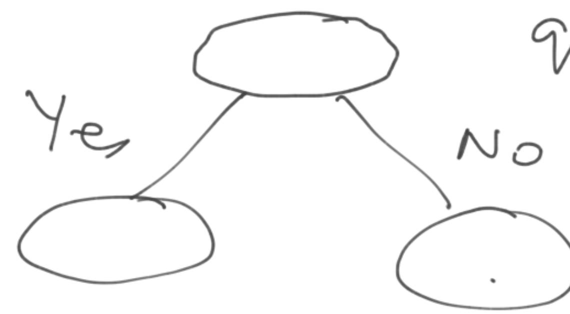
$(x_1, y_1) \ldots, (x_m, y_m)$

↑ classes. M classes

Getting optimal decision tree is NP complete

Heuristic algorithm

Greedy algorithm.



question

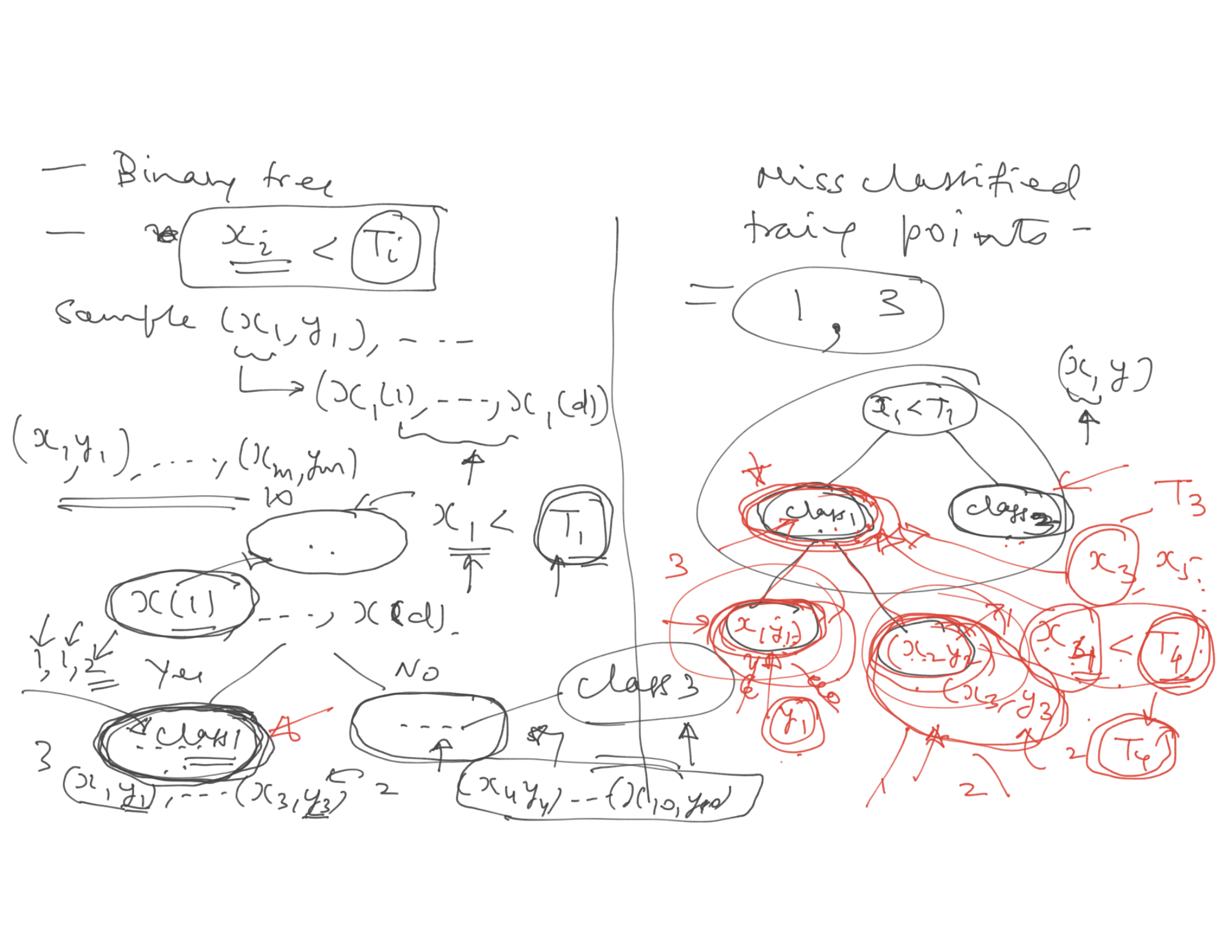
- what question

$x_t < T_t$

$x_1 + x_2^2 < 3.5$

$f(x_1, \ldots, x_d)$

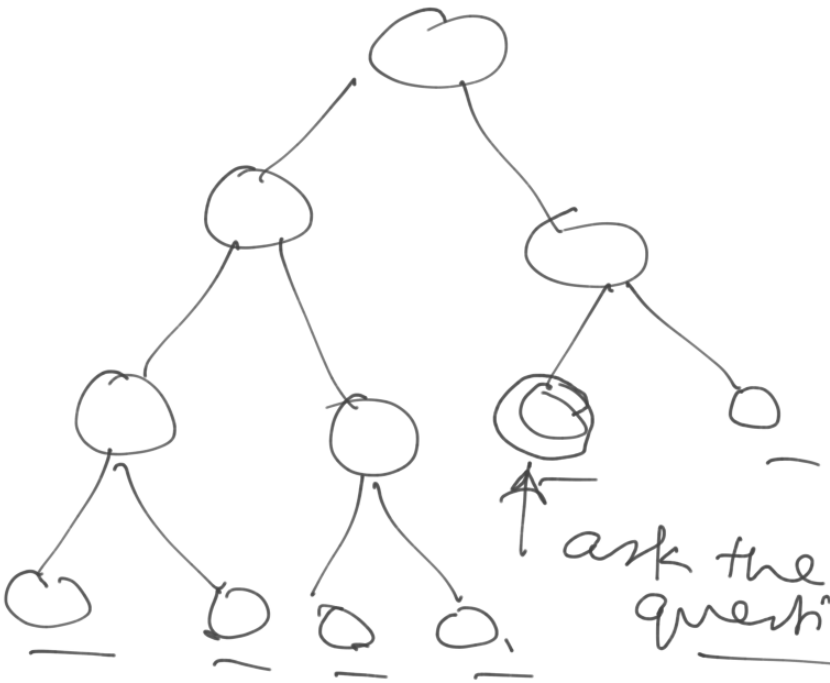
- Binary decision tree
- Questions will be $x_i < T_i$


— Binary tree
$x_i < T_i$
Sample $(x_1, y_1), \ldots$
$\hookrightarrow (x_1(1), \ldots, x_1(d))$
$(x_1, y_1), \ldots, (x_m, y_m)$
10
$x_1 <$ T_1
$x(1) \ldots x(d)$
1, 1, 2
Yes
No
class 1
class 3
3
$(x_1, y_1), \ldots (x_3, y_3)$
2
$(x_4 y_4) \ldots (x_{10}, y_{10})$
Misclassified train points =
1, 3
(x, y)
$x_1 < T_1$
class 1
class 3
T_3
x_3
x_5
3
$x_1 y_1$
$x_2 y_2$
$x_3 y_3$
$x_4 < T_4$
y_1
T_4
2
2

Greedy Algorithm

① Any leaf node will be labelled by the majority class.

② At any given node n_t we have to decide whether to divide it or not: if training sample is having large error then we may want to divide it



— Among leaf nodes choose node to divide which may have largest no errors

- After choosing next node to divide need to choose the question x_t

$x_t < T_t$

what threshold?

what feature?

- pick the feature and a good threshold
- Then change the feature & try with a good threshold ...

- Pick the best feature & threshold.

- The criterion chosen to decide among the different nodes & features is no of missclassified sample points.
- Prob of error

other ceiteria for selection of nodes and questions at a time :-

— Entropy

— Gini index.

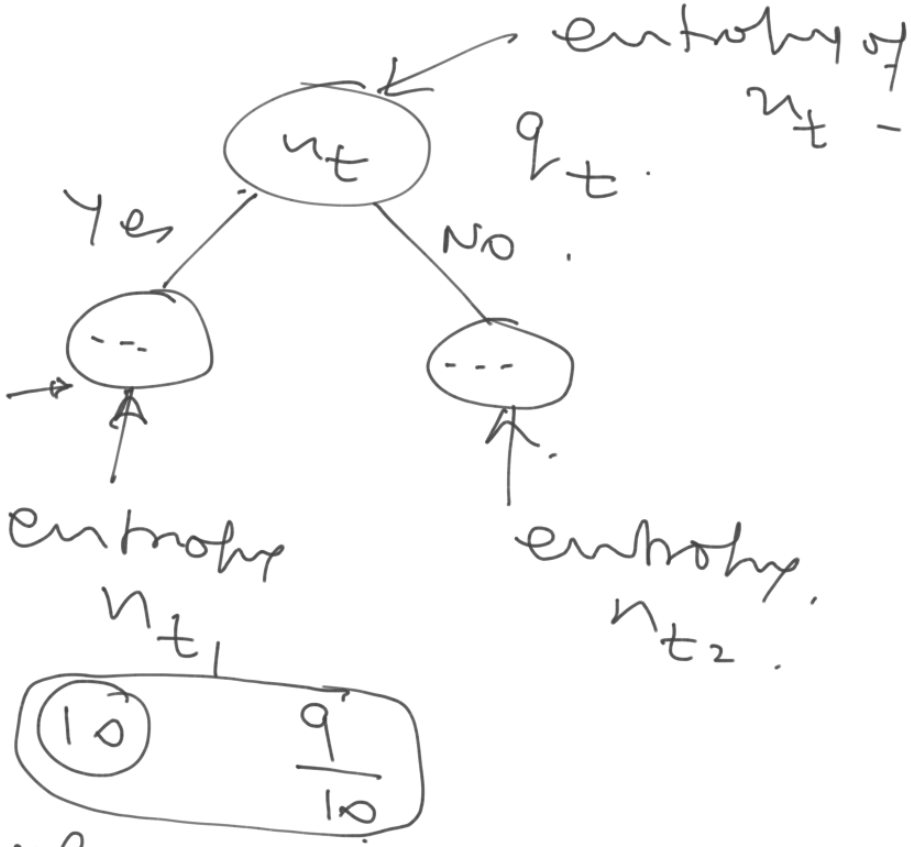
Given a node n_t

This node has m_t sample points and

$p_l = \frac{m_t(l)}{m_t}$ $\quad m_t(l)$ = no of sample points of class l in n_t.

entropy of n_t

$= -\sum_{l=1}^{k} p_l \log p_l$



Gini index :-

p_l = prob of class l

$$\sum_{l=1}^{K} p_l (1 - p_l).$$

entropy & Gini index are large when the node is $p_l = \frac{1}{K}$ $\forall l$.

and they are 0 if $p_l = 1$ for some l.
$= 0$ others.

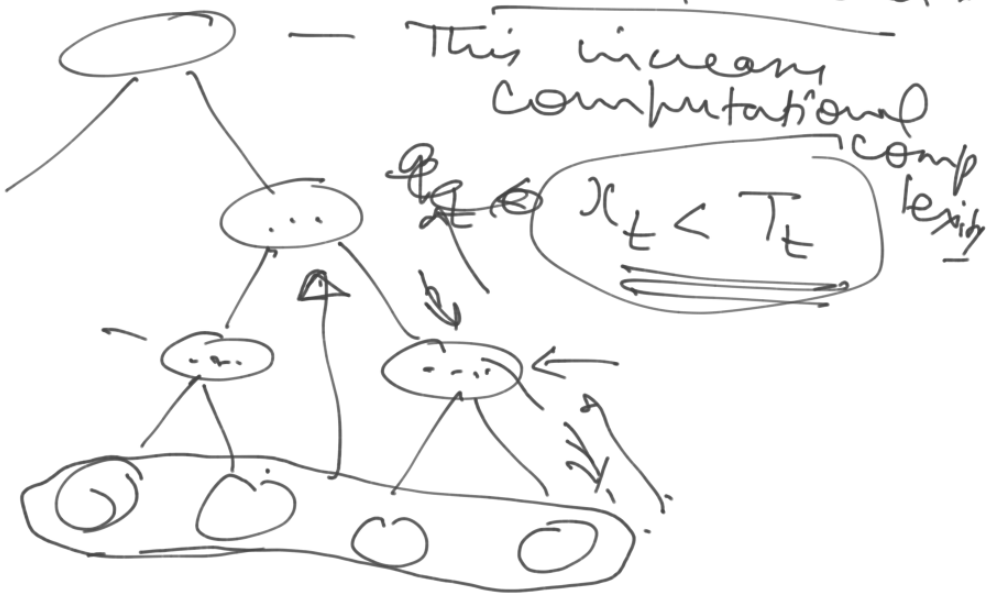
- All these three criteria are convex fns.
- But misclassification prob is not diff but entropy & Gini index are diff and also strictly convex
- entropy & Gini index are more commonly used methods/criteria.

impurity index of a node

- The algorithm is stopped when all the nodes have really low impurity or no of points at each node are quite low

Problems with greedy

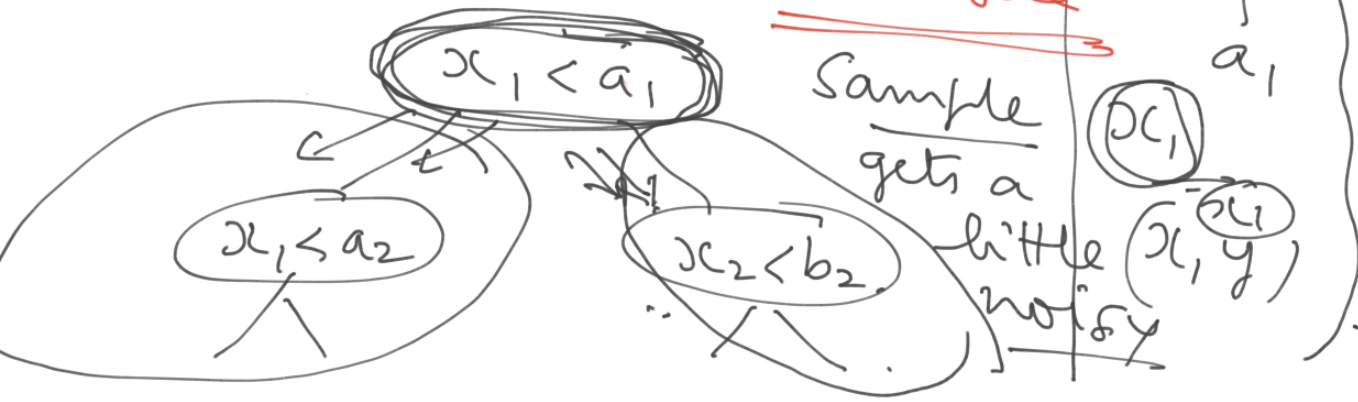
- It can give a bad split
  - This can be addressed by looking ahead.
  - This increases computational complexity



- It can give a large tree
  - This can do over fitting
- Grow and prune

Problem with Decision Tree

- These are unstable



Sample gets a little noisy

Random Forest

- Bunch of tree
- Then trees are trained in a random way.

S sample $(x_1, y_1), \ldots, (x_m, y_m)$

- Take a subsample S' randomly from S
- From the set of features $(x(1), \ldots, x(d))$ randomly pick d' feature $d' \leq \sqrt{d}$.
- Form a decision tree S' as features d'

using greedy.

This way we get Tree T_1

- Repeat the process
  - Take different subsample S' from S
  - take d' features randomly
  - Form Tree T_2 by greedy from (S', d')

$T_1, T_2, \ldots, T_\ell$ random tree

- For a new observation (OL) to classify :-
- Classify it using $T_1, T_2, \ldots, T_\ell$ Trees
- overall classification via majority decision
- Random Forest can provide as good performance as any.

- we have taken $d' \leq \sqrt{d}$.
  This way the correlation between different trees are low.
- Theoretical analysis is hard. Very limited results available
- One reason is the use of greedy algorithm.
- Some studies on performance are astrip instead of using greedy algorithm for tree randomly choose the node

---

- Using binary classification algorithms.
- OVA (One vs All.)
  K class $S = (x_1, y_1), \ldots (x_m, y_m)$.
  For each class $l \in Y$.
  make classifier h_l taking two class class1 → class l
  class2 → The rest class.

- h_l is formed from any of the binary classification algorithms.
- We have k classifier.
- For classifier l form scoring function f_l
  - suppose we used kernel method for binary classification

$$f_l(x) = \underbrace{y\left(w^T \cdot \Phi(x)\right)}_{y \in \{1, -1\}}$$

- $f_l(x) = p(x \mid \text{class } l)$

- pick the class with largest $f_l(x)$
- calibration problem $f_l(x)$
- This OVA works quite well in practice

OVO method :

OVO
↓
One vs One

- For each $l, l' \in \{1, \ldots, K\}$ make a binary classifier $h_{ll'}$ using subsample of S which have labels l and l'
- Now finally when a new measurement x comes
- $K(K-1)$ binary classifiers.
- classify x via all these classifiers and choose class with majority.
- Complexity of learning OVA is same as that of other multiclass classifiers
- But complexity of OVO can be very large

— But complexity of OVO still may not be very large because for $h_{ll'}$ the sample size to train from can be quite small.

— One problem with OVO is that for some $h_{ll'}$ the sample to train from can be quite small.

— But it does not have the calibration problem of OVA.

— If generalization error of $h_{ll'} \leq r$ $\forall\ l, l'$ then generalization error of overall classifier $\leq (k-1)r$.

— kNN and NN.

↳ k nearest nbr.

↳ Neural networks