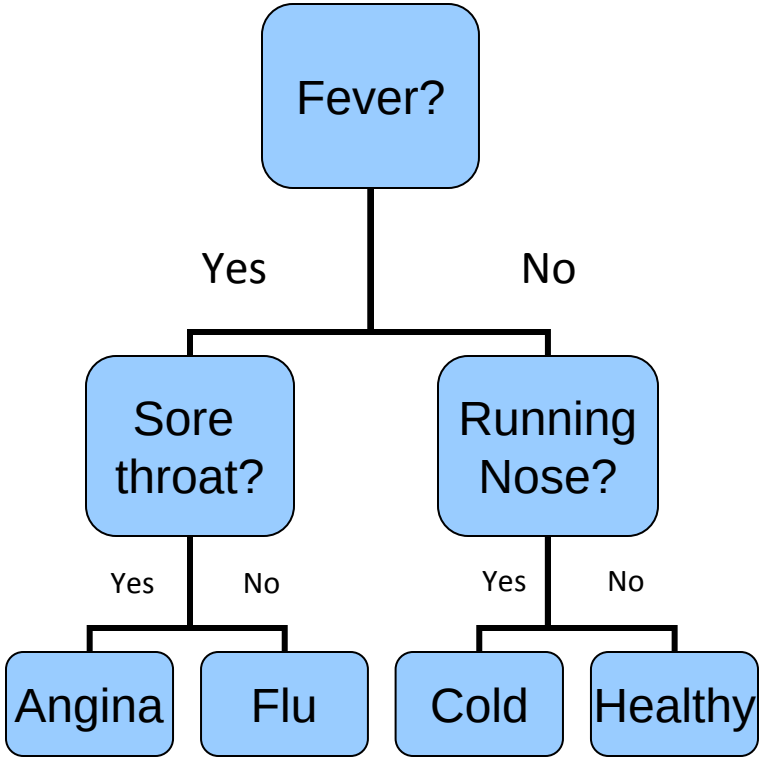
Oblique Trees Classification

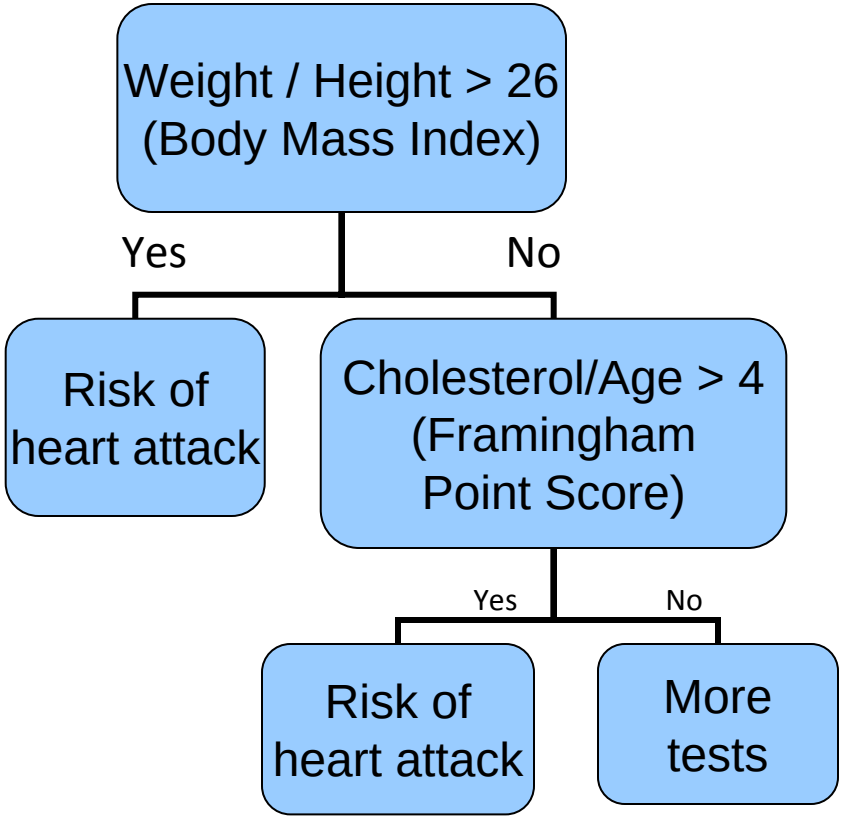
Andrejus Parfionovas

Adele Cutler

Utah State University

Tree-based classification:





Classical tree-based methods:

1. Pick a variable (1 dimension)
2. Find the best split to separate classes (Gini index as a measure of purity)
3. Repeat steps 1-2 and choose the best separable variable
4. Go to step 1 with fewer observations, until all nodes in the tree are pure

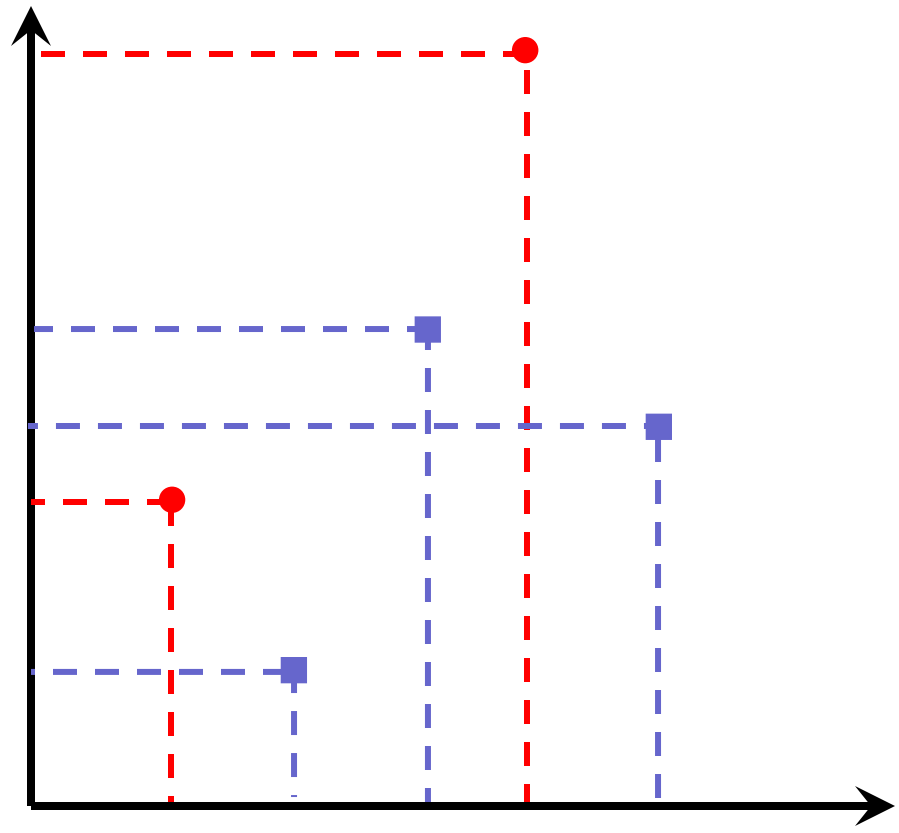
The feature space is now partitioned into a set of rectangles

Newly proposed algorithm:

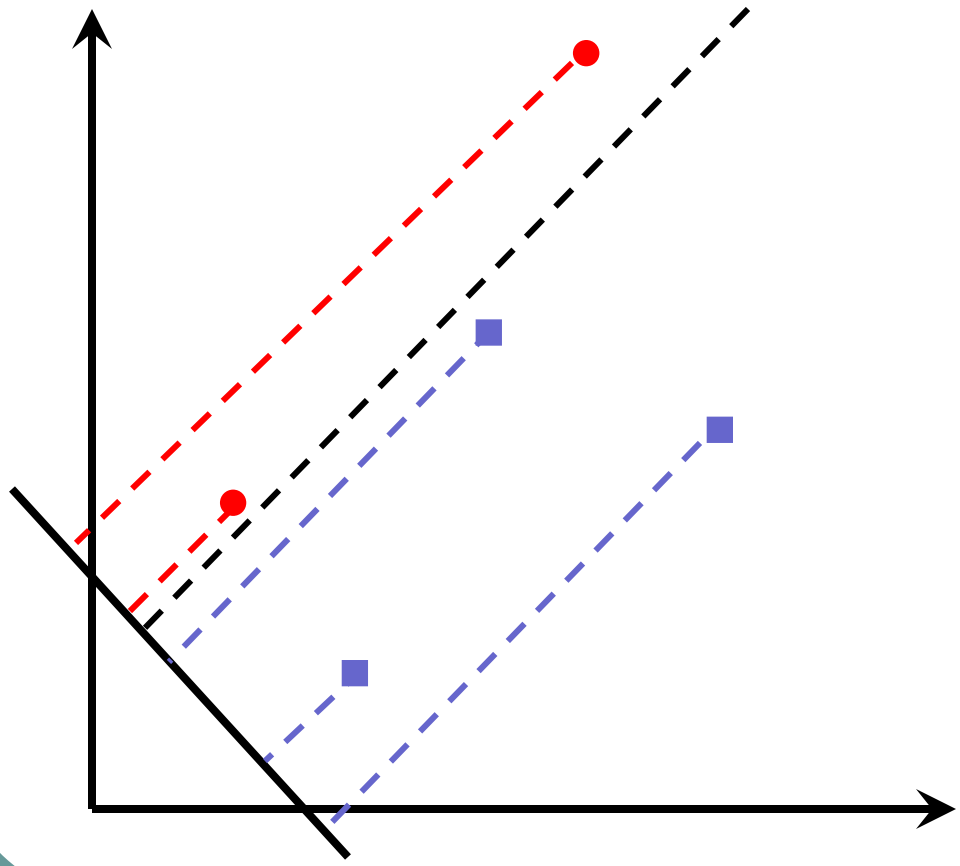
1. Pick **two** variables
2. Find the best split in 2D space to separate classes (Gini-based criteria)
3. Repeat steps 1-2 and choose the best separable **pair** of variables
4. Go to step 1 with fewer observations, until all nodes in the tree are pure

The feature space is now partitioned by oblique linear splits

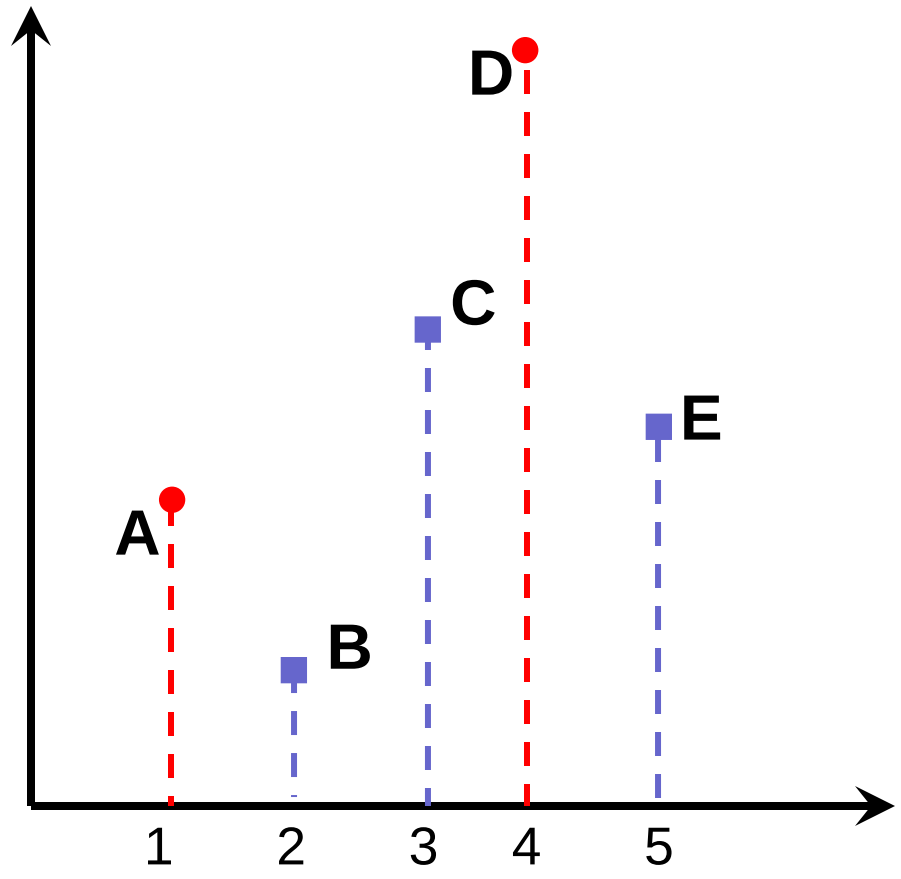
Problem formulation:



Finding the best split in 2D:



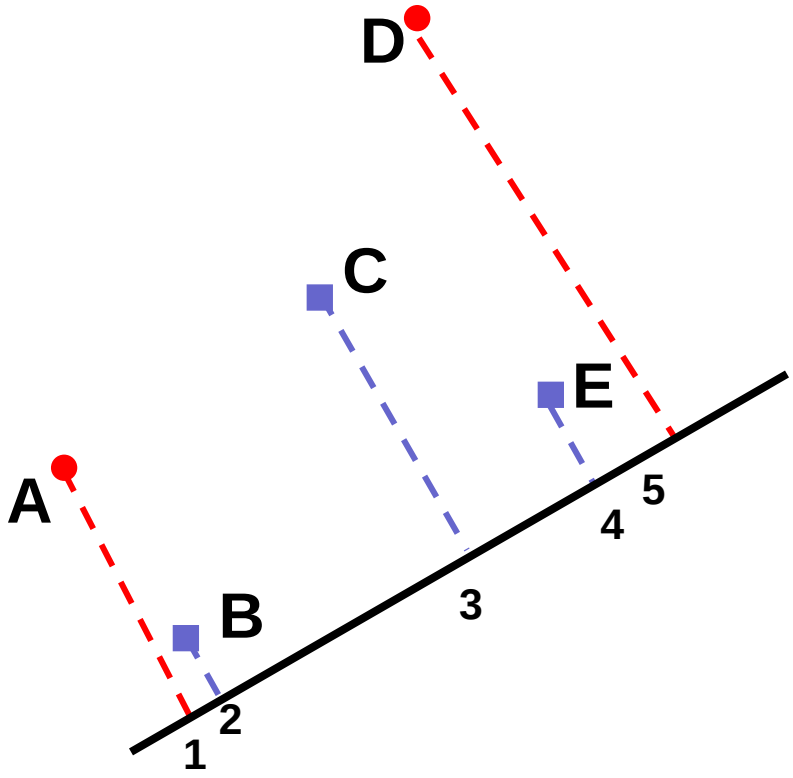
Finding the best split in 2D:



Slopes

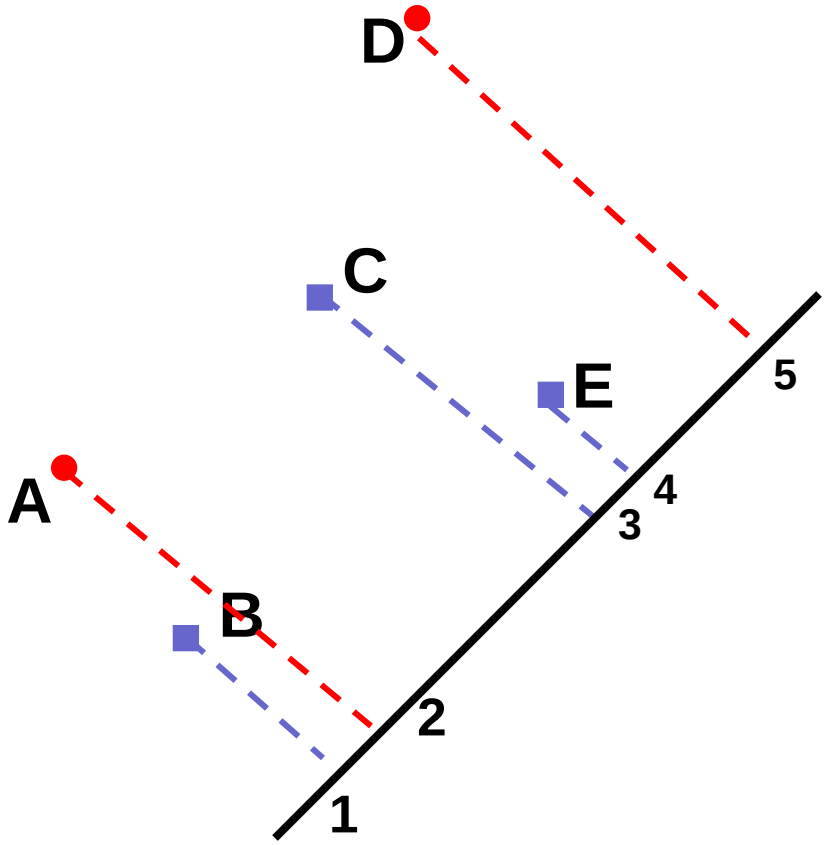
DE	-3
AB	-1
CE	-.5
AE	.25
BE	.67
AC	1
AD	1.33
CD	2
BD	2.5
BC	3

Finding the best split in 2D:



Slopes	
DE	**-3**
AB	-1
CE	-.5
AE	.25
BE	.67
AC	1
AD	1.33
CD	2
BD	2.5
BC	3

Finding the best split in 2D:



Slopes

DE	-3
AB	**-1**
CE	-.5
AE	.25
BE	.67
AC	1
AD	1.33
CD	2
BD	2.5
BC	3

Finding the best split in 2D:

Ordering of the points

A	.3	B	.47	C	.47	D	.4	E
A		B		C		E	.3	D
B	.3	A		C		E		D
B		A		E	same	C		D
B		E	.26	A		C		D
E	same	B		A		C		D
E		B		C	0	A		D

Slopes

DE	-3
AB	-1
CE	-.5
AE	.25
BE	.67
AC	1
AD	1.33
CD	2
BD	2.5
BC	3

Algorithm Performance:

$O(n \cdot n_1 - n_1^2) \sim$ between $O(n - 1)$ and $O(n^2)$,

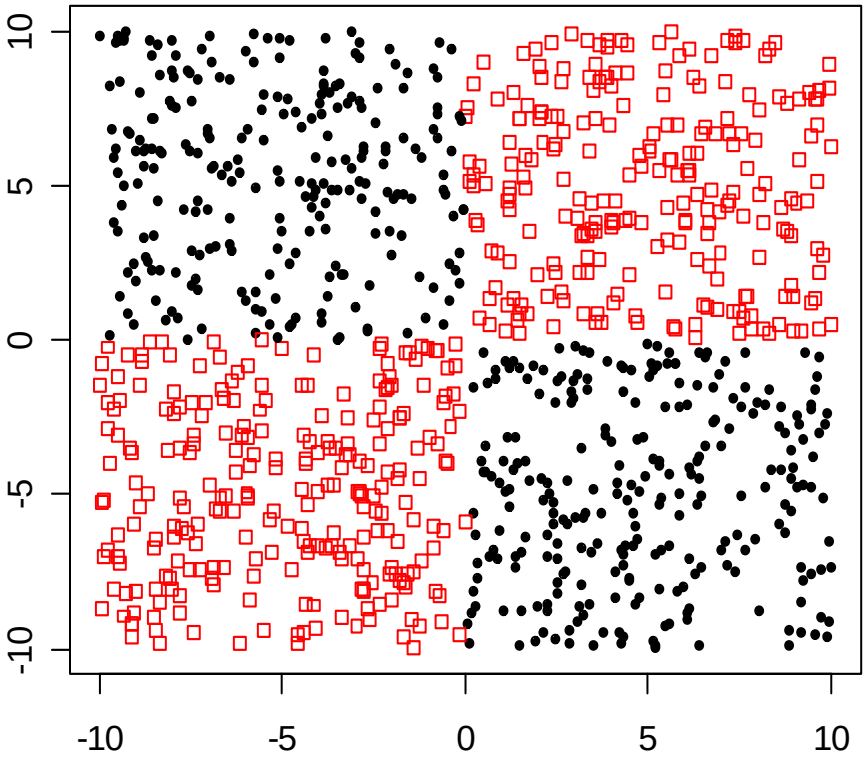
where n – sample size,

$n_1 \in (0, n)$ – biggest class size

No need to recalculate Gini index!

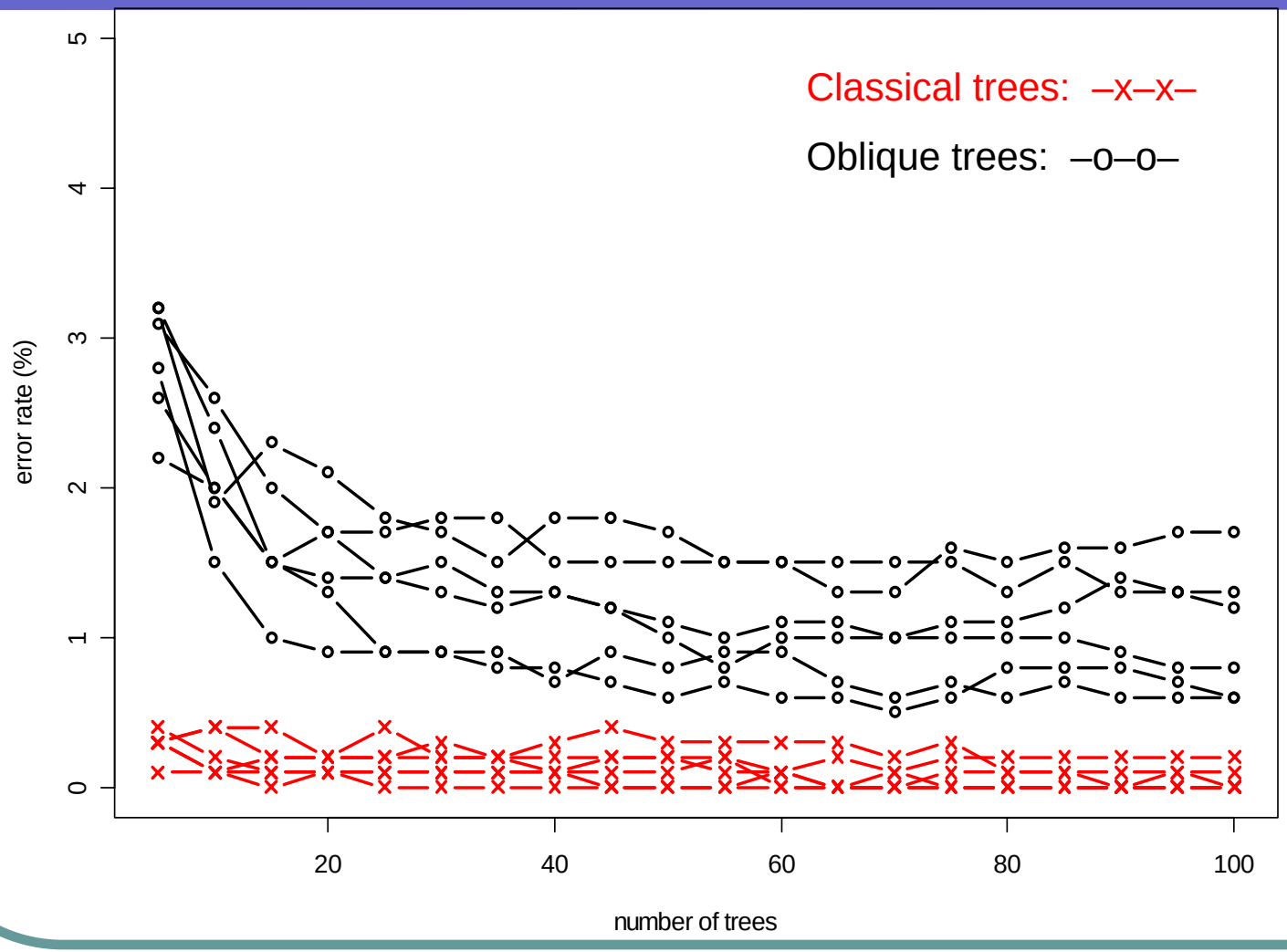
(can easily be updated)

Simulation: Orthogonal XOR

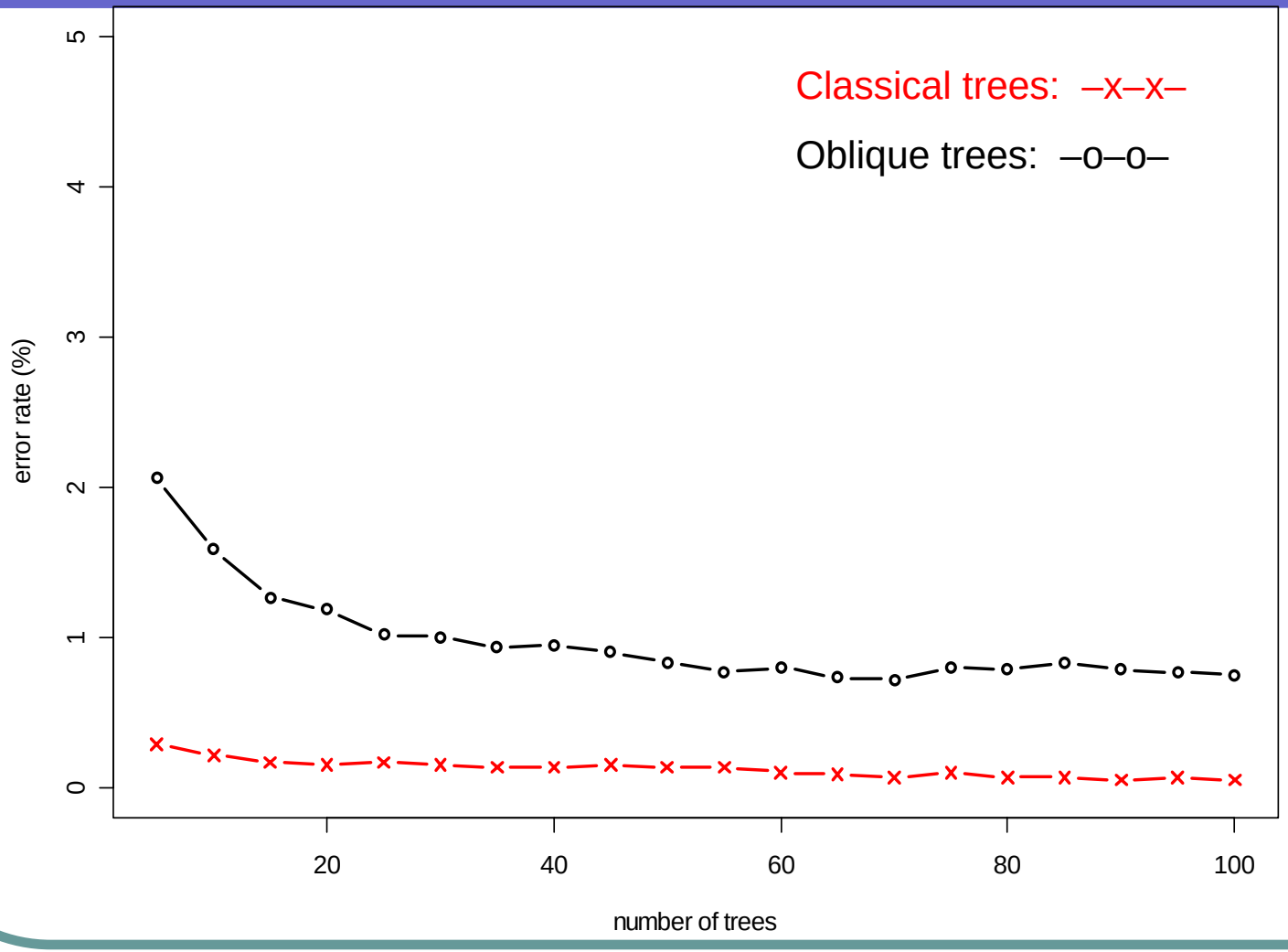


- Training sample = 500
- Test sample = 1000
- # of repetitions = 6

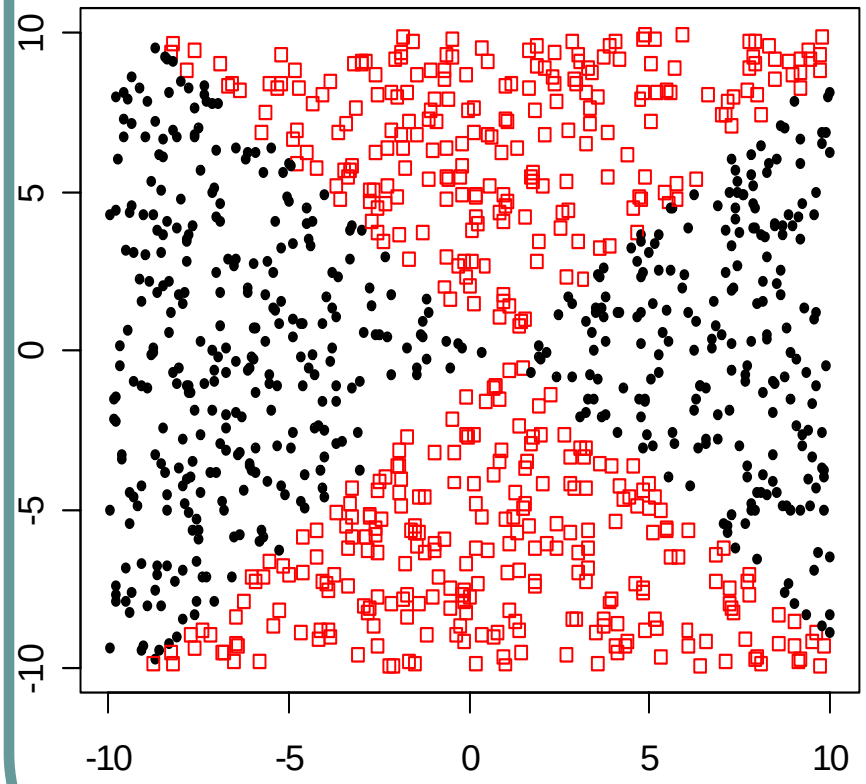
Simulation: Orthogonal XOR



Simulation: Orthogonal XOR

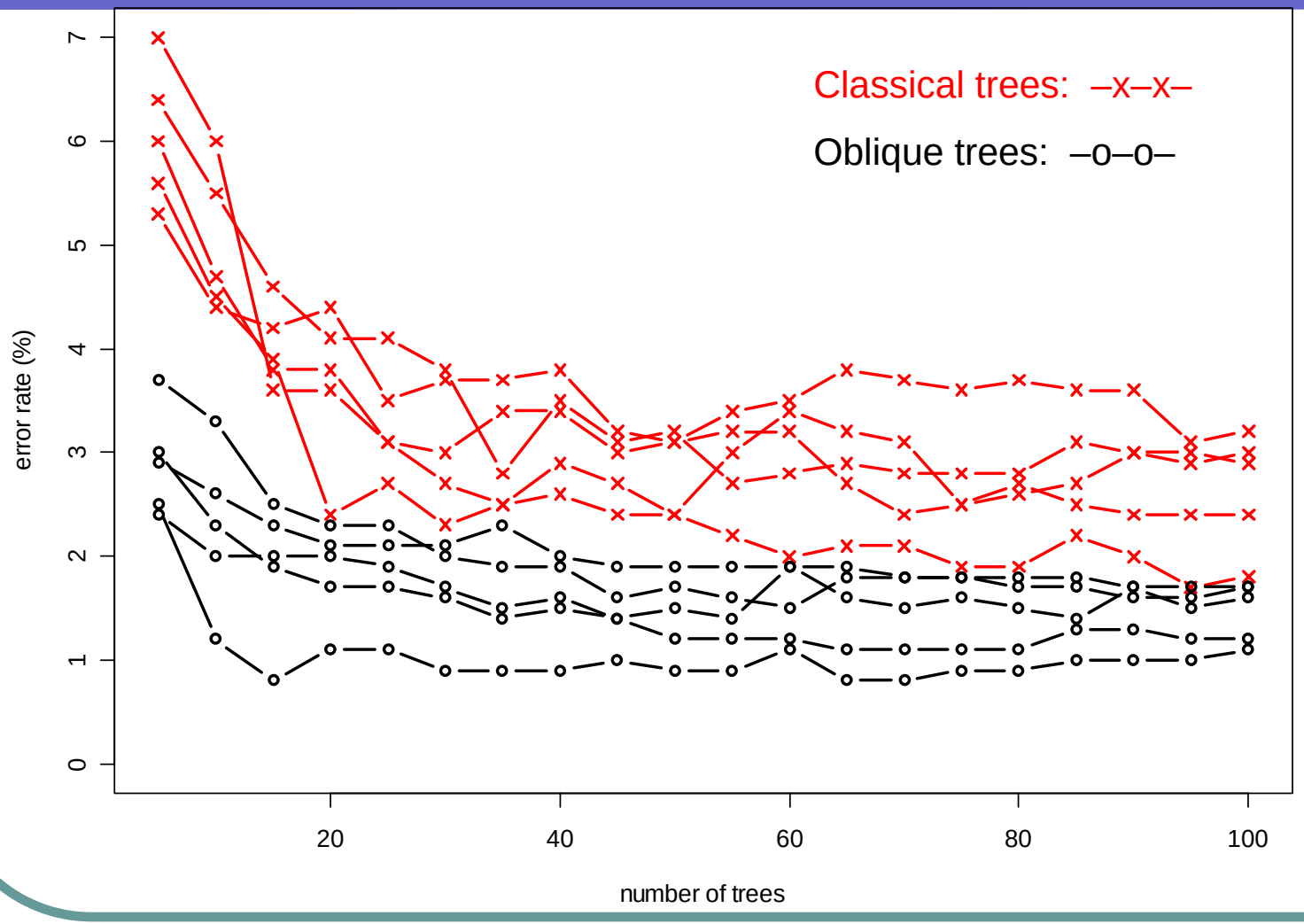


Simulation: Diagonal XOR

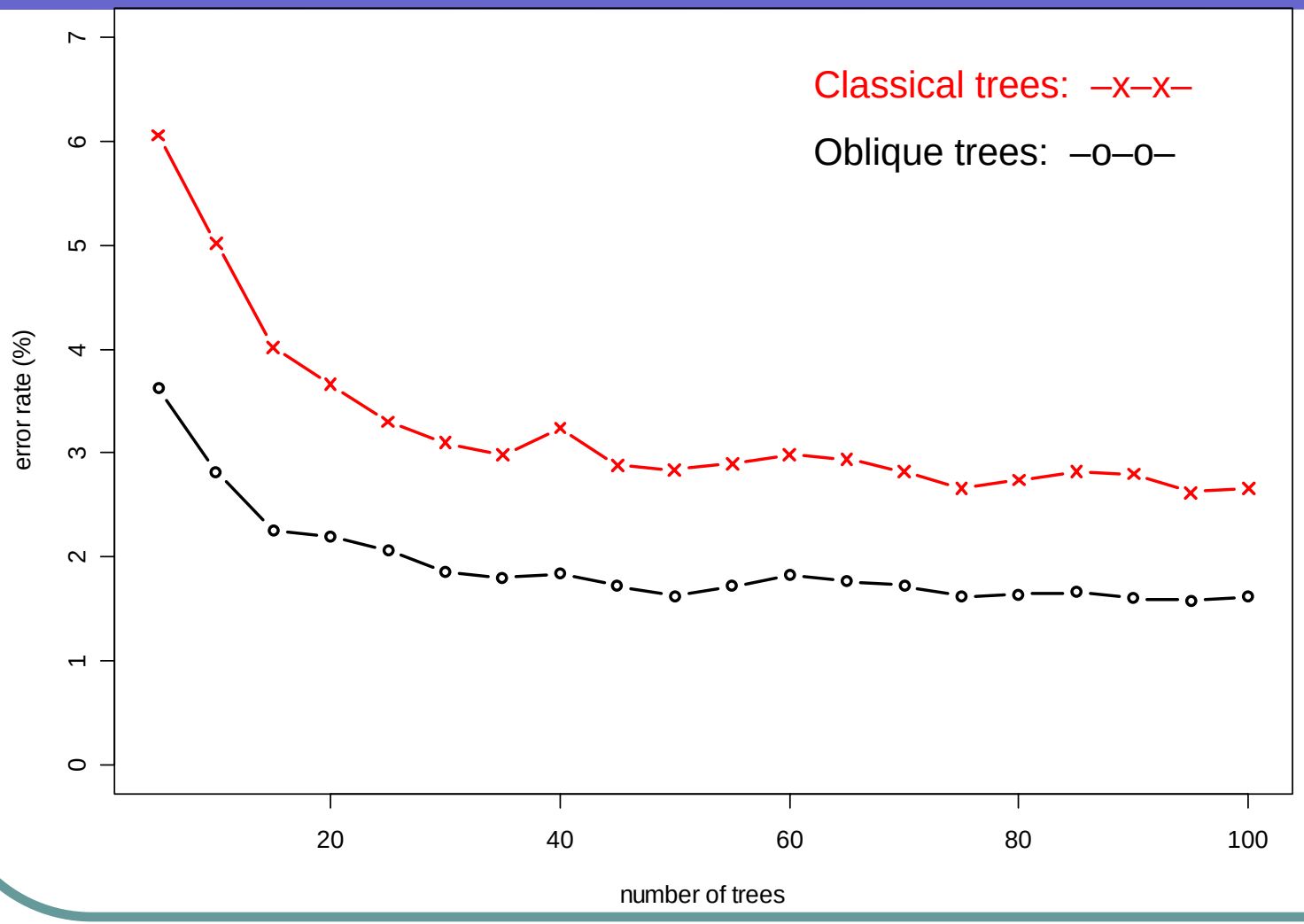


- Training sample = 400
- Test sample = 1000
- # of repetitions = 5

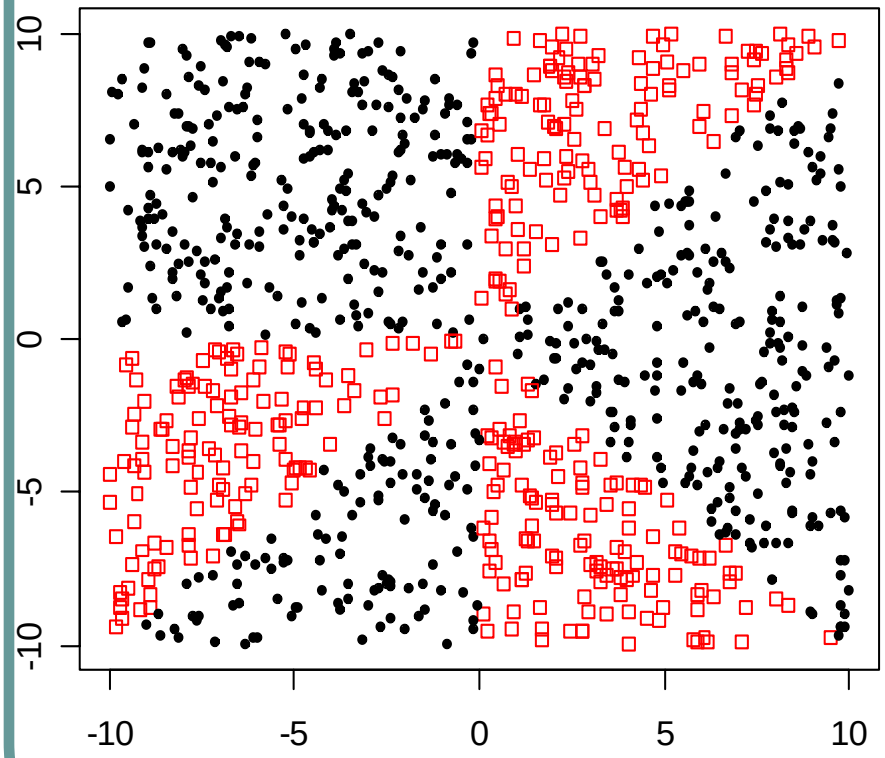
Simulation: Diagonal XOR



Simulation: Diagonal XOR

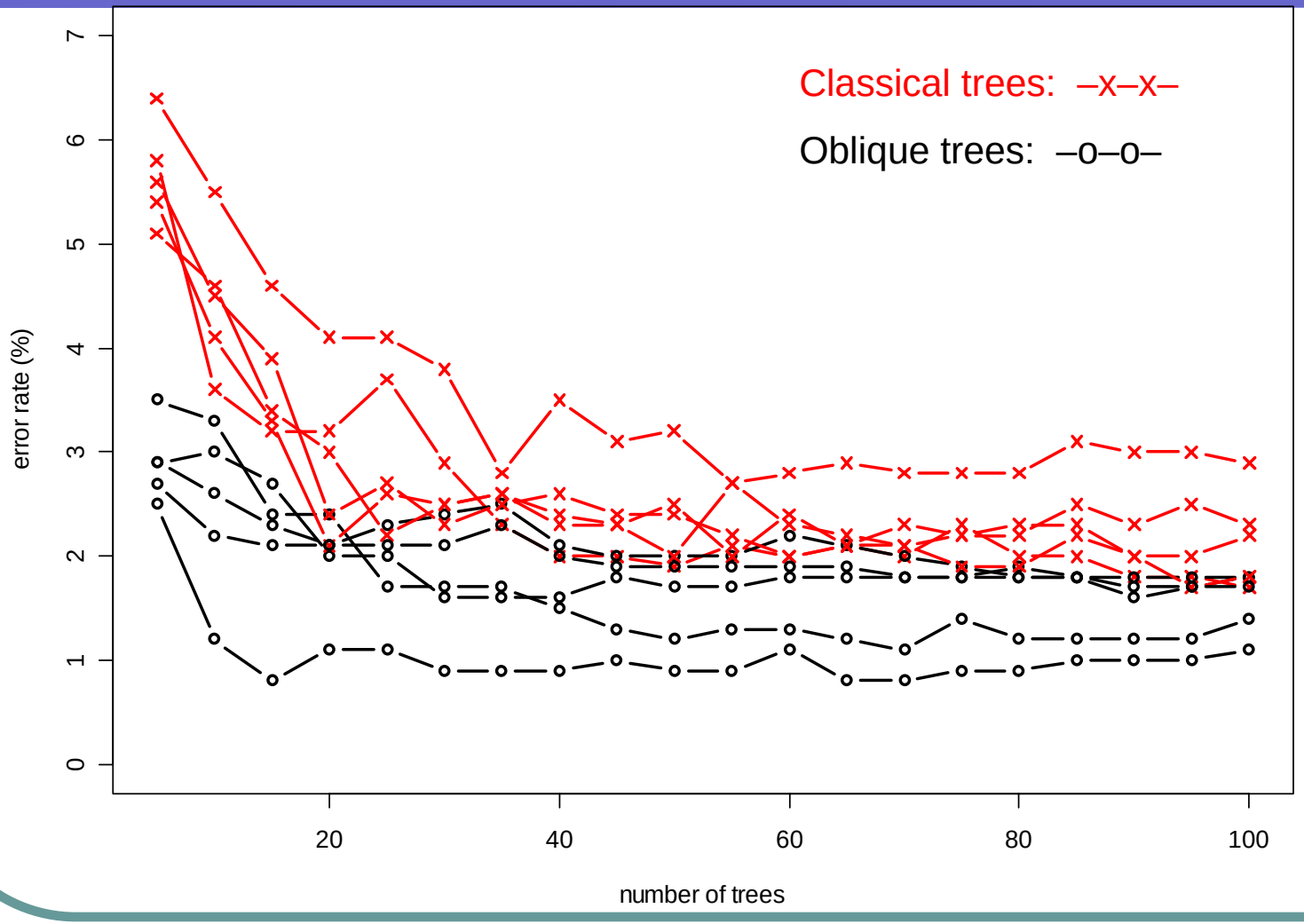


Simulation: Mixed XOR

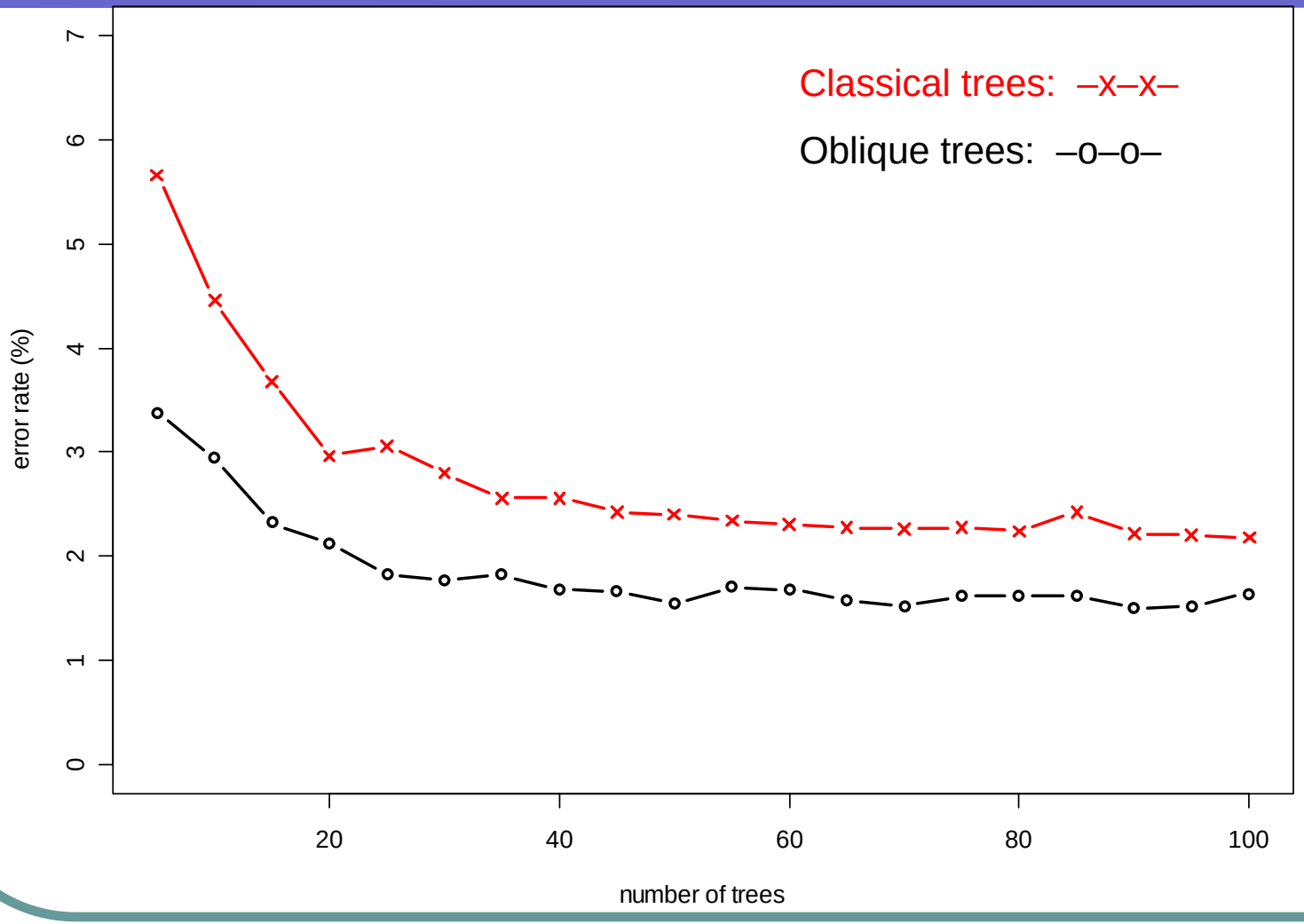


- Training sample = 400
- Test sample = 1000
- # of repetitions = 5

Simulation: Mixed XOR



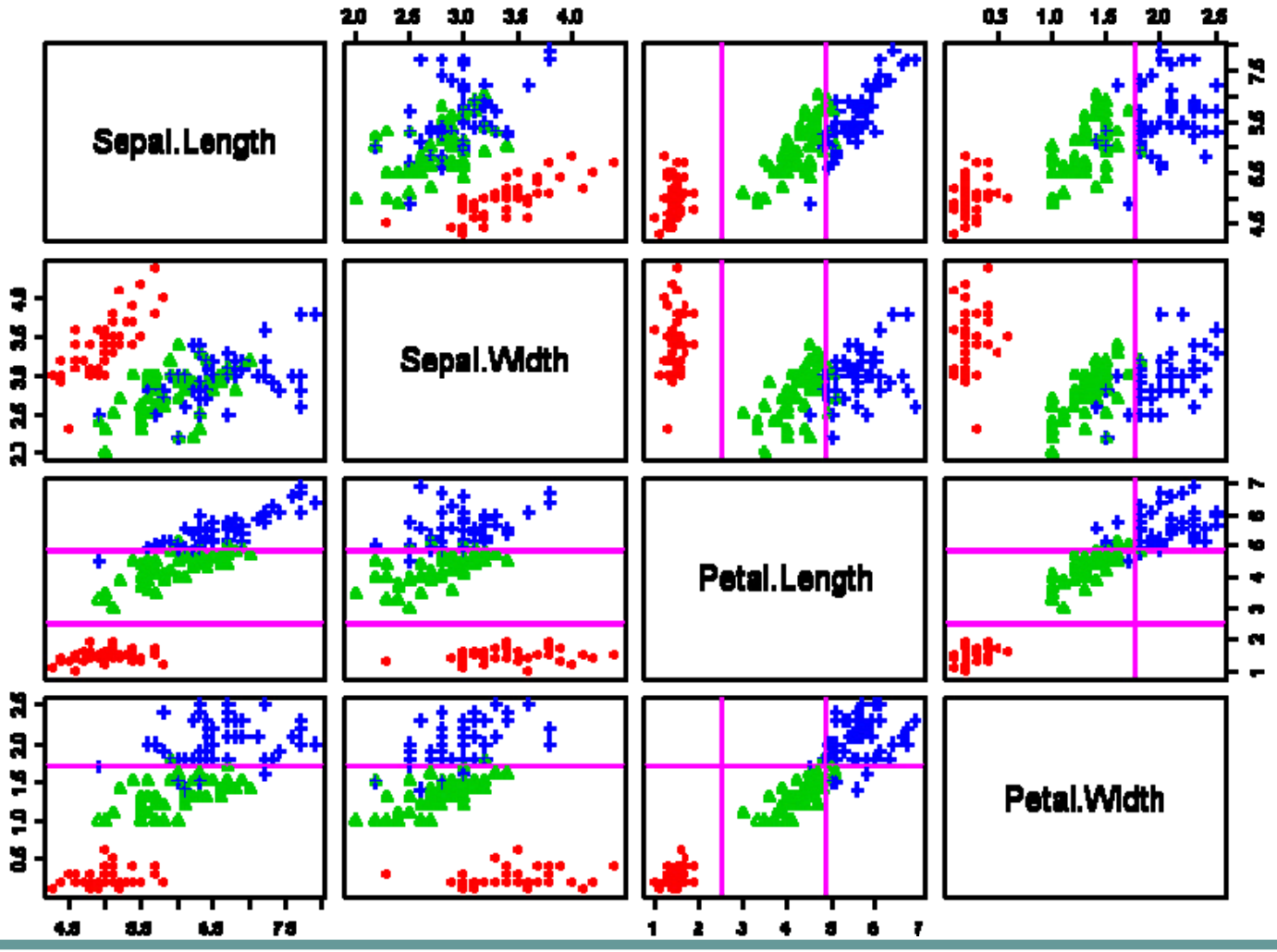
Simulation: Mixed XOR



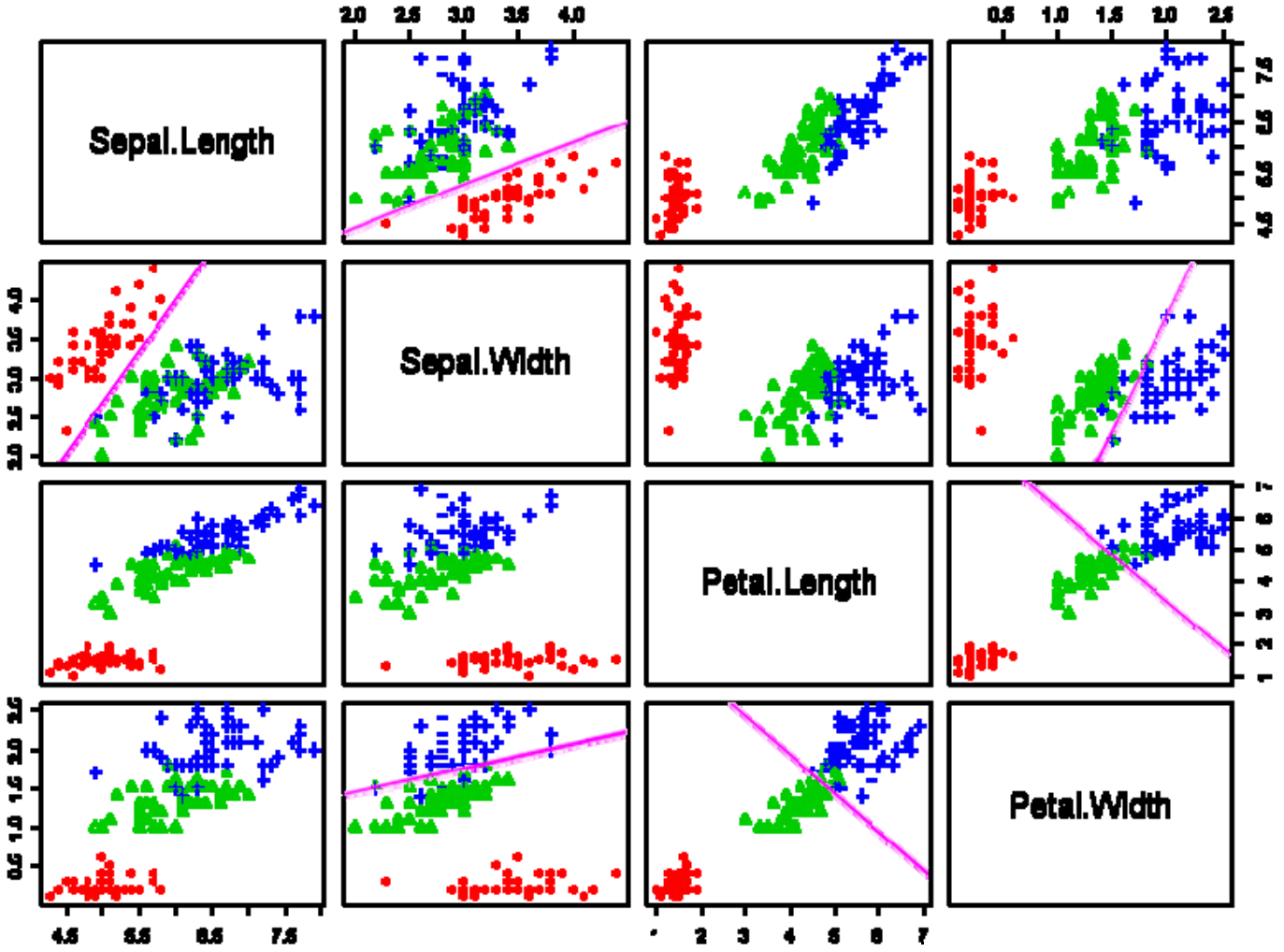
Example: Anderson's Iris Data

- 150 observations (50 in each class)
- 3 classes (species)
- 4 variables:
  - Sepal Length
  - Sepal Width
  - Petal Length
  - Petal Width

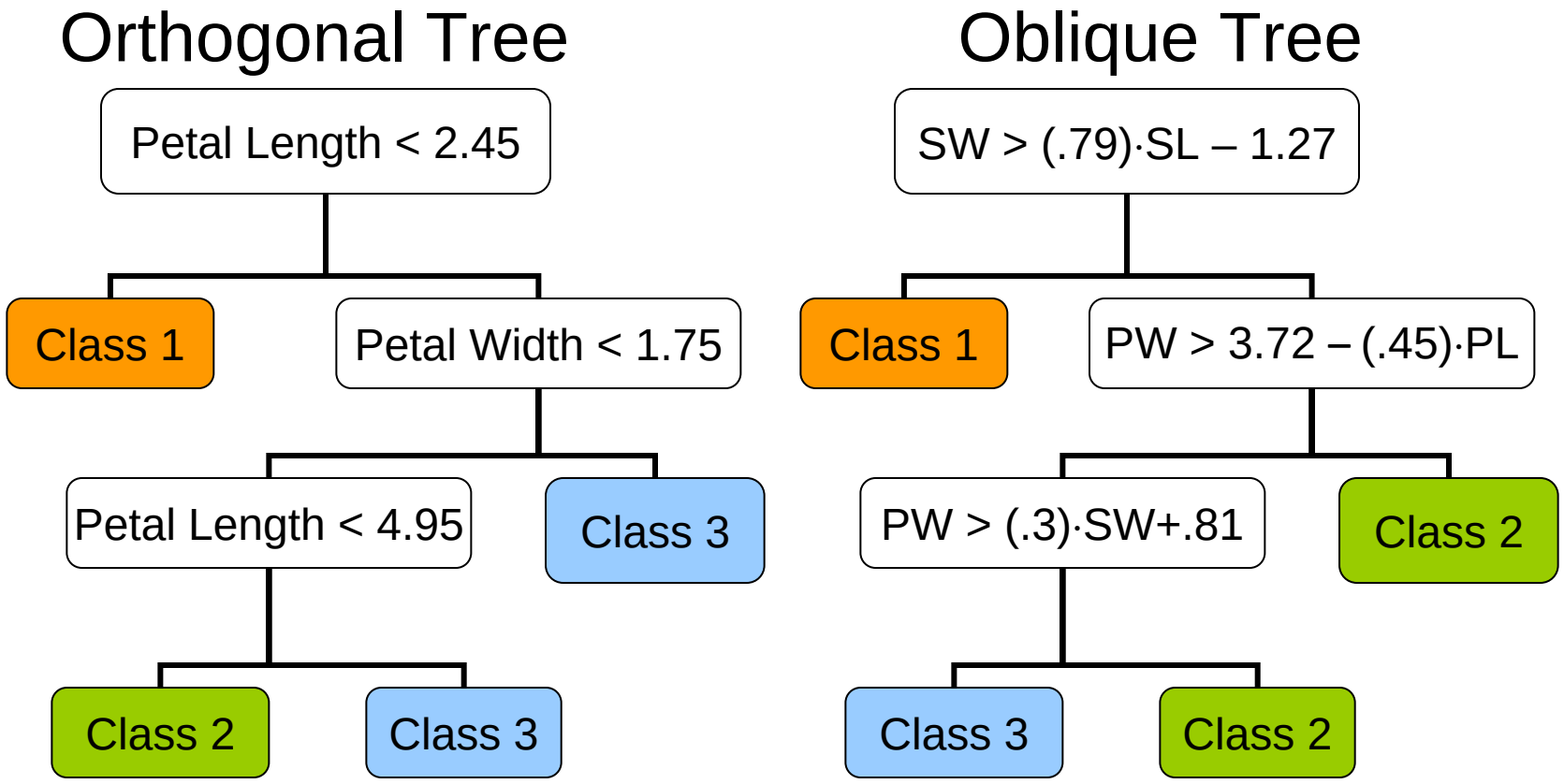
Classical (orthogonal) tree:



Oblique tree classification:



Trees comparison:



Summary:

- ☑ Consistently, smaller error rate
- ☑ Fewer trees required for desired accuracy
- ☑ Smoother, more adaptive class separation
- ☑ Provides insights of the data structure (variables selection)

- ☒ Orthogonal separation – stay with classics

Future investigations:

- Randomization
  - When choosing between ties
  - Try random pairs of variables
- High dimensional data
- C++ based package for R users

Thank you for your attention!

Special thanks to Dr. Adele Cutler
for her contribution and advice

More information:

http://cc.usu.edu/~andrej/
andrej@cc.usu.edu
adele.cutler@usu.edu