Random Decision Forests

Tin Kam Ho
AT&T Bell Laboratories
600 Mountain Avenue, 2C-548C
Murray Hill, NJ 07974, USA

TAAD, 04-fev-2022

Jorge Silva

Problem

- Traditional Decision Trees can’t grow much in complexity without sacrificing accuracy
- When complexity is needed, the model is suboptimal

Problem

- Traditional Decision Trees can’t grow much in complexity without sacrificing accuracy

- When complexity is needed, the model is suboptimal

Solution proposal: Use a stochastic method to improve accuracy for arbitrary complexities

Oblique Decision Trees

- Binary decision trees use parallel hyperplanes (linear function of features)

- Oblique Decision Trees are more general and often smaller

Oblique Decision Trees

- Two methods
  - Central Axis Projection **(CAP)**
    - Use one hyperplane to separate classes clusters
    - Calculate classes average
    - Draw a line between them (central axis)
    - Project the data
    - Search along the line with a fixed step and calculate error
  - Perceptron training **(PER)**
    - Use a fixed step and Euclidian distance to find one hyperplane for each non terminal node

Systematic Creation of Multiple Trees

- Both methods induce *bias* when choosing the planes
- On the other hand, using multiple classifiers has shown better results then using just one

With those in mind, they propose combining results from trees in random subspaces of the feature space (2^n).

Results with handwritten digits

- Testing the theory with handwritten digits

- Two different feature vectors were used
  - $\boldsymbol{f_1}$ = 20x20 pixels
  - $\boldsymbol{f_2}$ = 20x20 pixels plus horizontal, vertical and diagonal neighborhoods

Results with handwritten digits

1. Perceptron achieves the same accuracy as the Central Projection using smaller trees

Table 1: Number of Terminal Nodes and Classification Accuracies for Each Class

	CAP(f_1)		CAP(f_2)		PER(f_1)		PER(f_2)	
class	#nodes	%corr	#nodes	%corr	#nodes	%corr	#nodes	%corr
0	108	95.24	93	96.54	9	91.77	2	96.21
1	87	98.27	80	98.45	12	97.32	3	98.53
2	226	87.13	167	91.94	18	83.30	5	93.22
3	247	88.88	212	90.90	17	85.24	6	92.72
4	183	89.99	139	90.20	14	83.92	5	92.65
5	251	86.07	193	90.25	15	78.67	3	92.39
6	121	91.09	110	89.72	12	87.07	4	93.14
7	185	89.95	161	90.34	16	87.98	5	93.10
8	288	83.01	234	84.57	22	79.39	6	89.36
9	279	84.98	239	87.44	19	83.55	5	91.22
all	1975	89.57	1628	91.11	154	86.01	44	93.32

Results with handwritten digits

2. Accuracy is better using a couple of the 2^n possible spaces, and using more trees

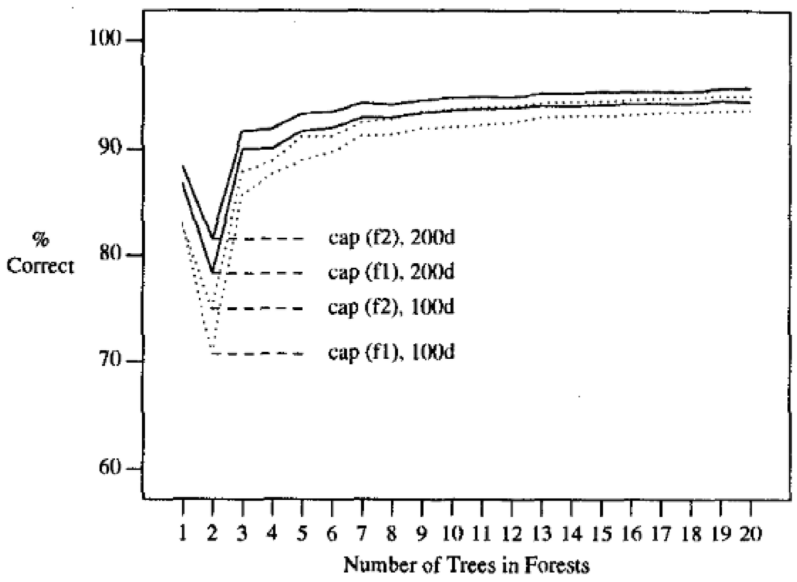

Figure 1: Classification Accuracy (% correct) of Forests Constructed by Central Axis Projection (in 100- and 200- dimensional random subspaces)

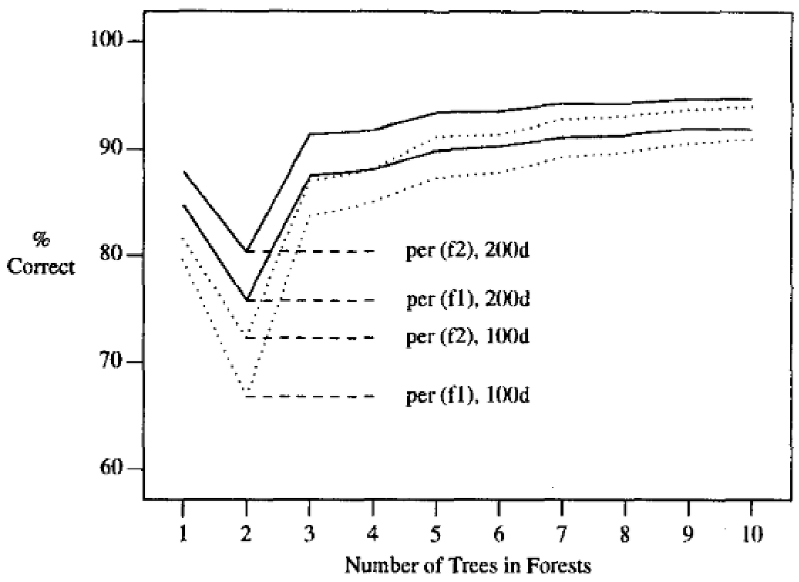

Figure 2: Classification Accuracy (% correct) of Forests Constructed by Perceptron Training Algorithm (in 100- and 200- dimensional random subspaces)