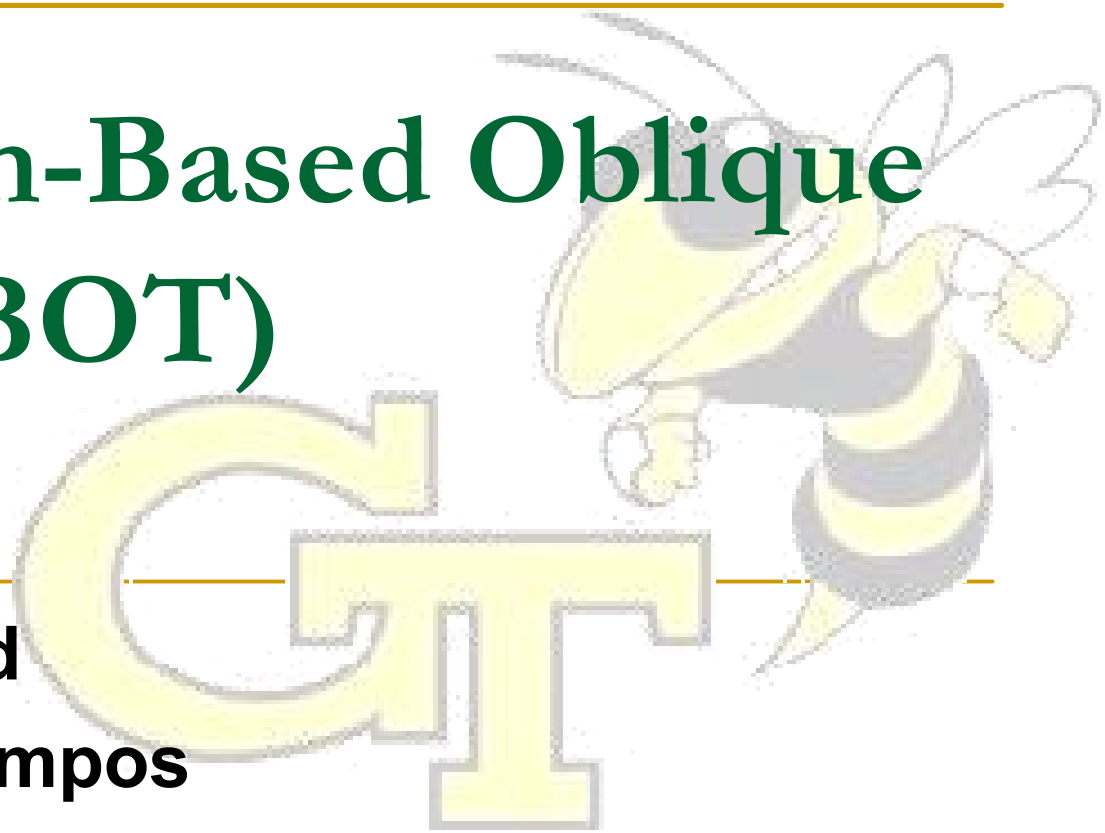
Perceptron-Based Oblique Trees (P-BOT)



Ben Axelrod

Stephen Campos

John Envarli

Problems with Traditional Classifiers

- Decision Trees
  - Axis-parallel splits only
  - Limited representational power
- Neural Networks
  - Slow for non-linear fits
  - Suffer from overfitting

Our Solution: Perceptron-Based Oblique Tree (P-BOT)

- Use both in one
  - Implement a decision tree
  - Implement a linear neural net at each node
  - Combines the strengths and addresses the weaknesses of both algorithms
- Results:
  - For some datasets, more accurate than both
  - For others, faster than NN's and more accurate than DT's

P-BOT in more detail

- Infogain calculations for all attributes (DT)
- Neural net at each node
  - No hidden layer
  - # output nodes = # of classes
  - Normalized features
  - Back propagation using classification error

P-BOT in more detail (cont.)

- Infogain calculation for neural network
  - Classification error used as estimation of infogain
- Best infogain selected as the split condition
  - DT splits simplify dataset
  - Neural splits allow linear combination of features
- Hypothesis space
  - Sets of non-intersecting convex polytopes that cover the entire feature space

High Level Illustration by Example

- Dataset
  - Three classes
  - 2 continuous attributes
  - 1 discrete attribute
    - Input node per value
- Neural net split at root
- Discrete split at left child

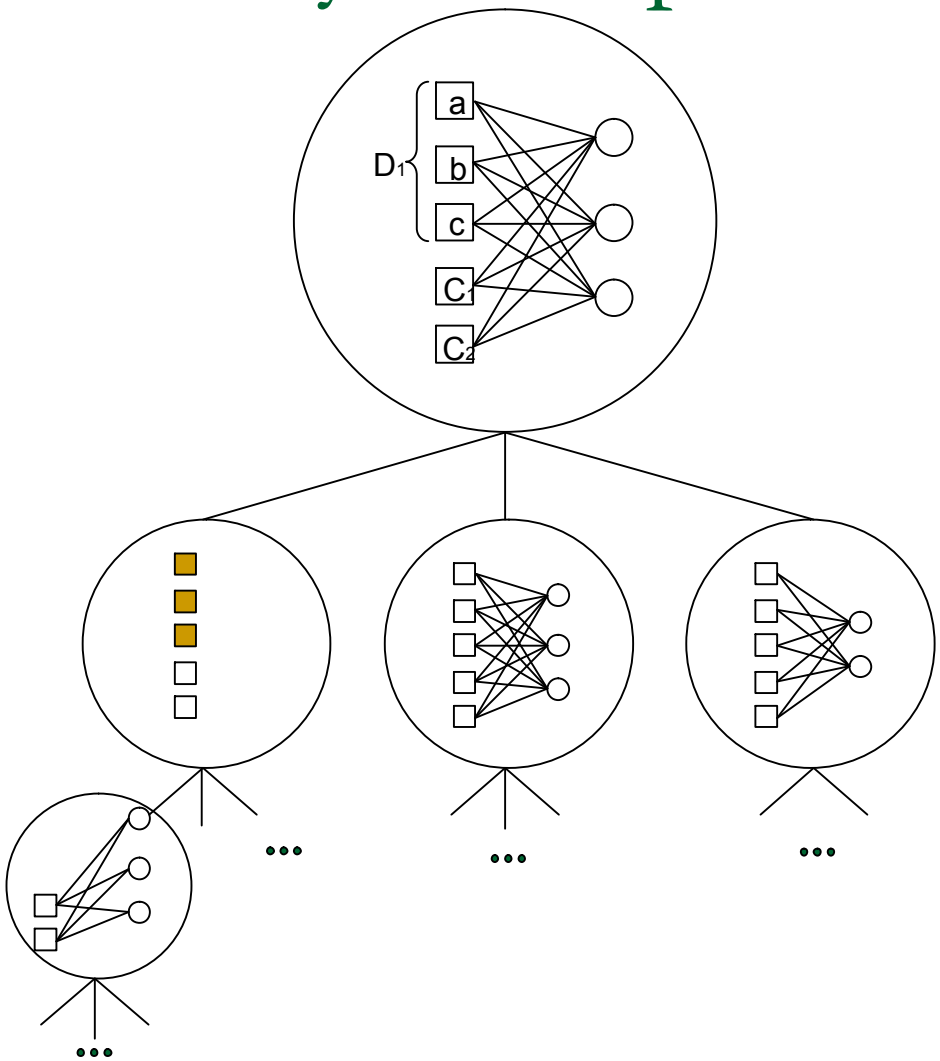
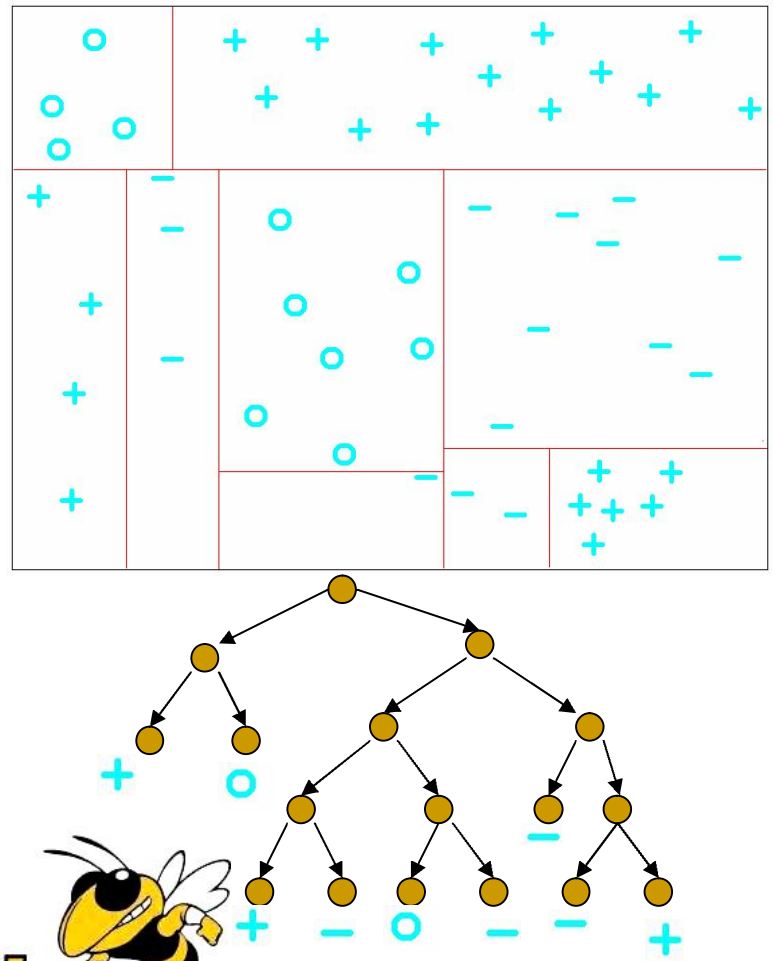
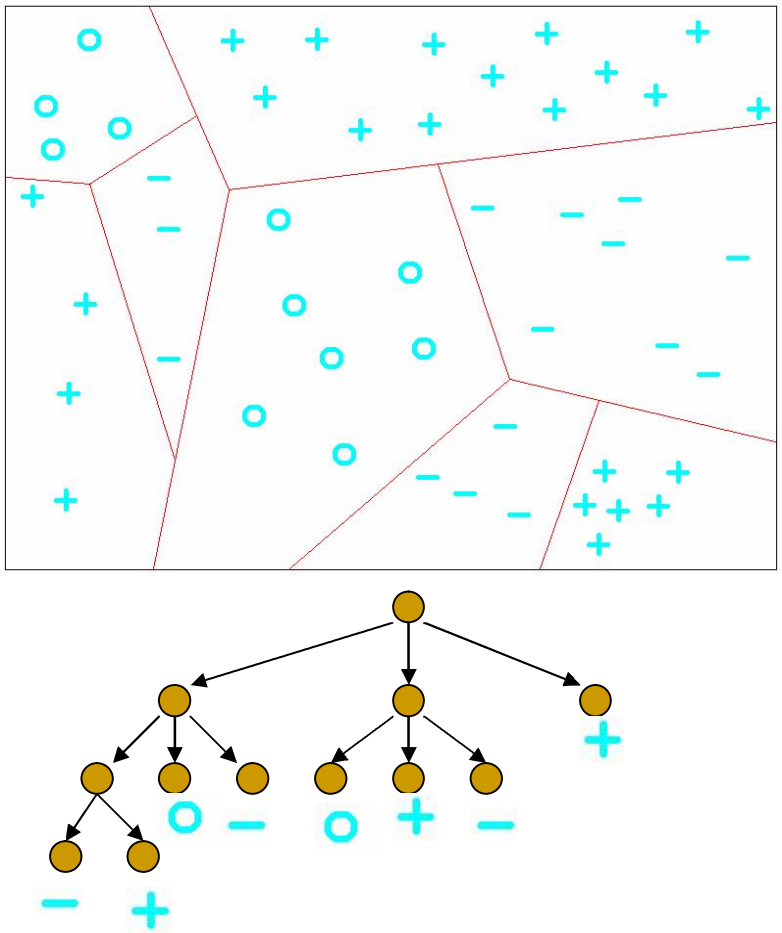


Illustration on a 2D Dataset

Decision Tree

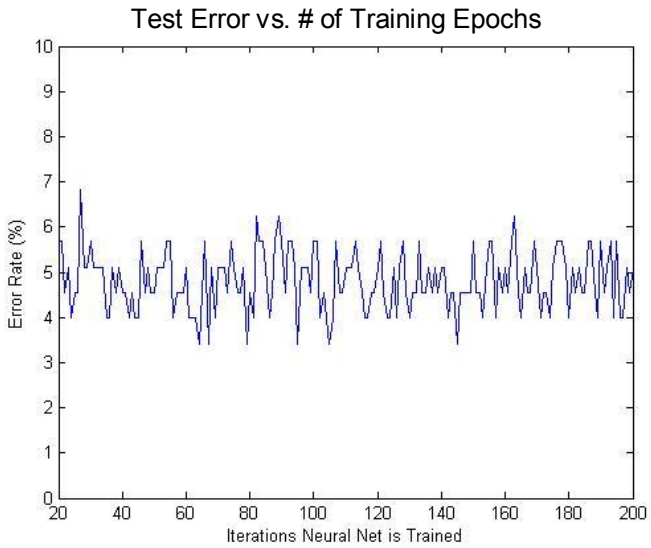


P-BOT



Avoiding Overfitting

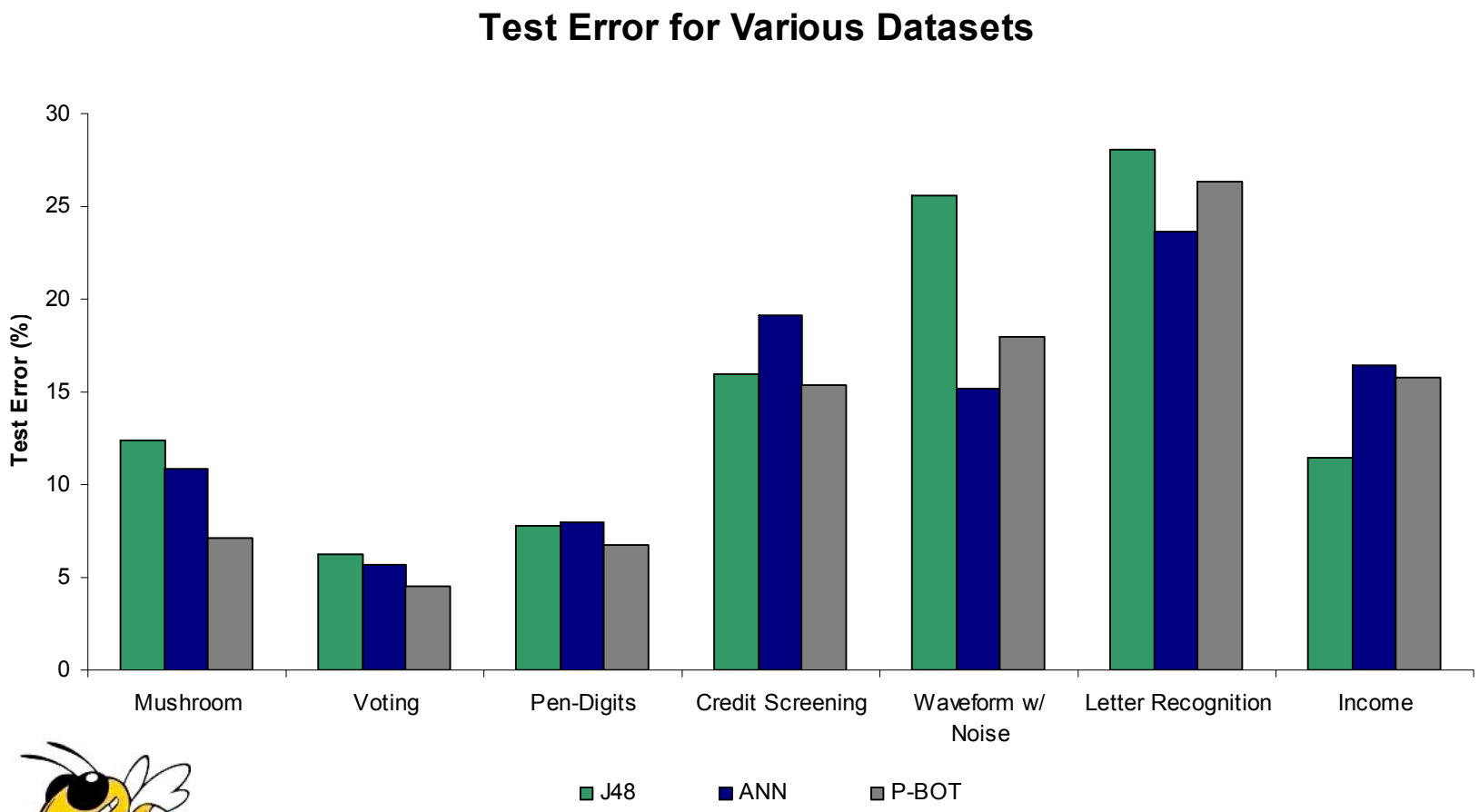
- Neural net – High number of epochs cannot result in overfitting (line fit)
- Tree – Pruning over validation set



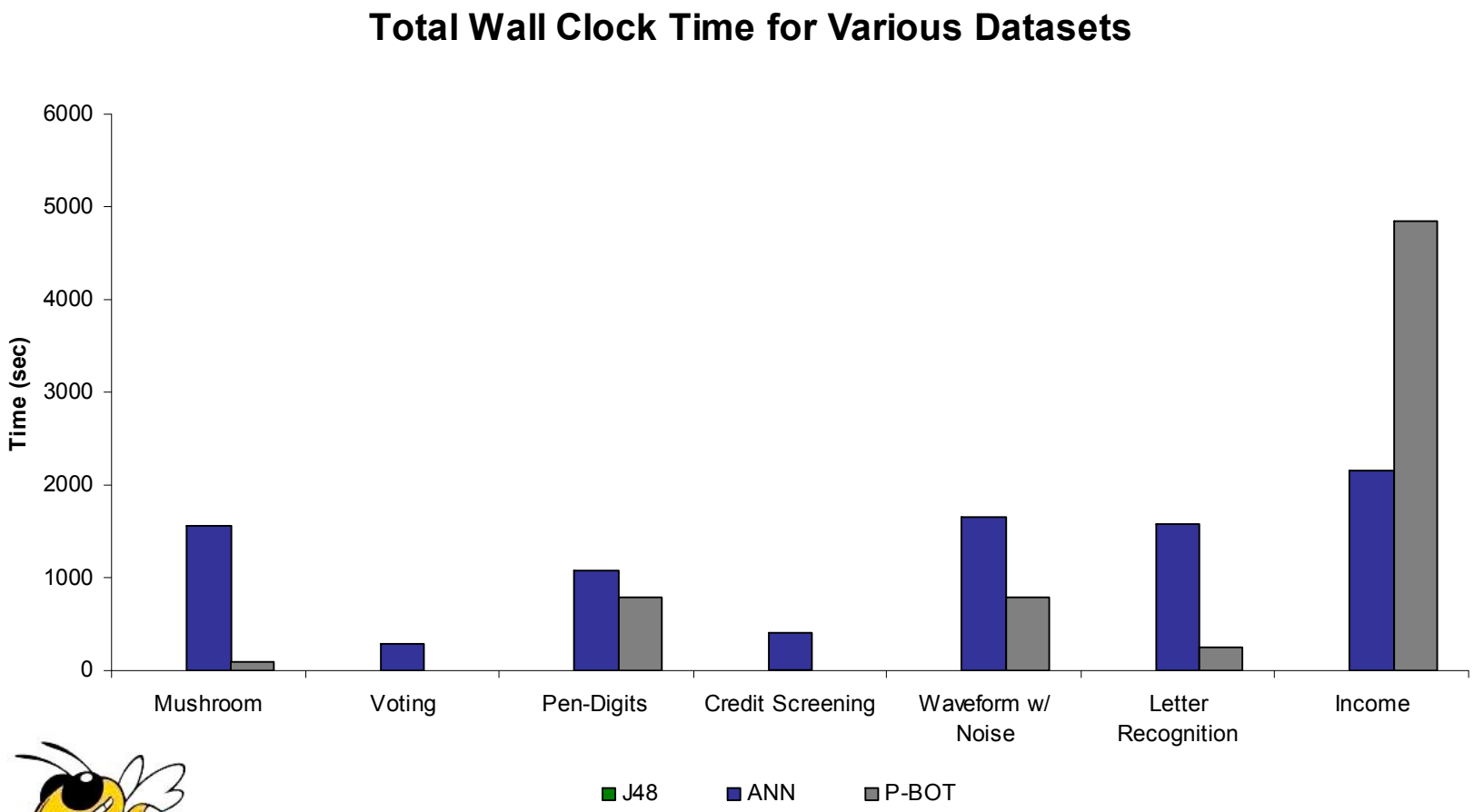
Related Work

- Oblique Decision Tree - OC1
  - Single hyperplane at each node
  - Perturbation algorithm to construct hyperplane
- Linear Machine Decision Trees
  - Very similar to P-BOT
  - Pays more attention to terminating the gradient search
- Neither approach considers attribute splits

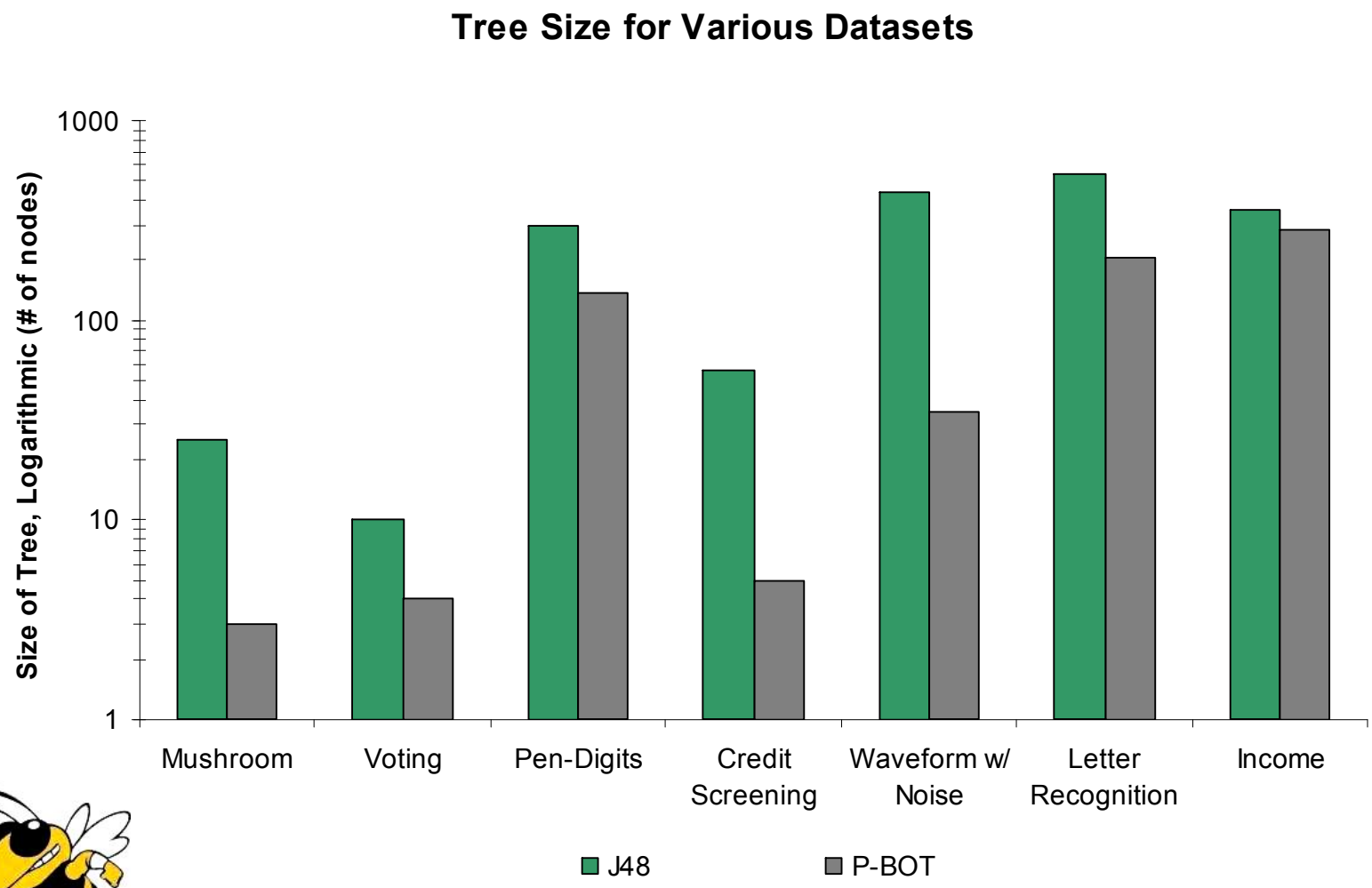
Results


Test Error for Various Datasets
Test Error (%)
30
25
20
15
10
5
0
Mushroom
Voting
Pen-Digits
Credit Screening
Waveform w/ Noise
Letter Recognition
Income
J48
ANN
P-BOT

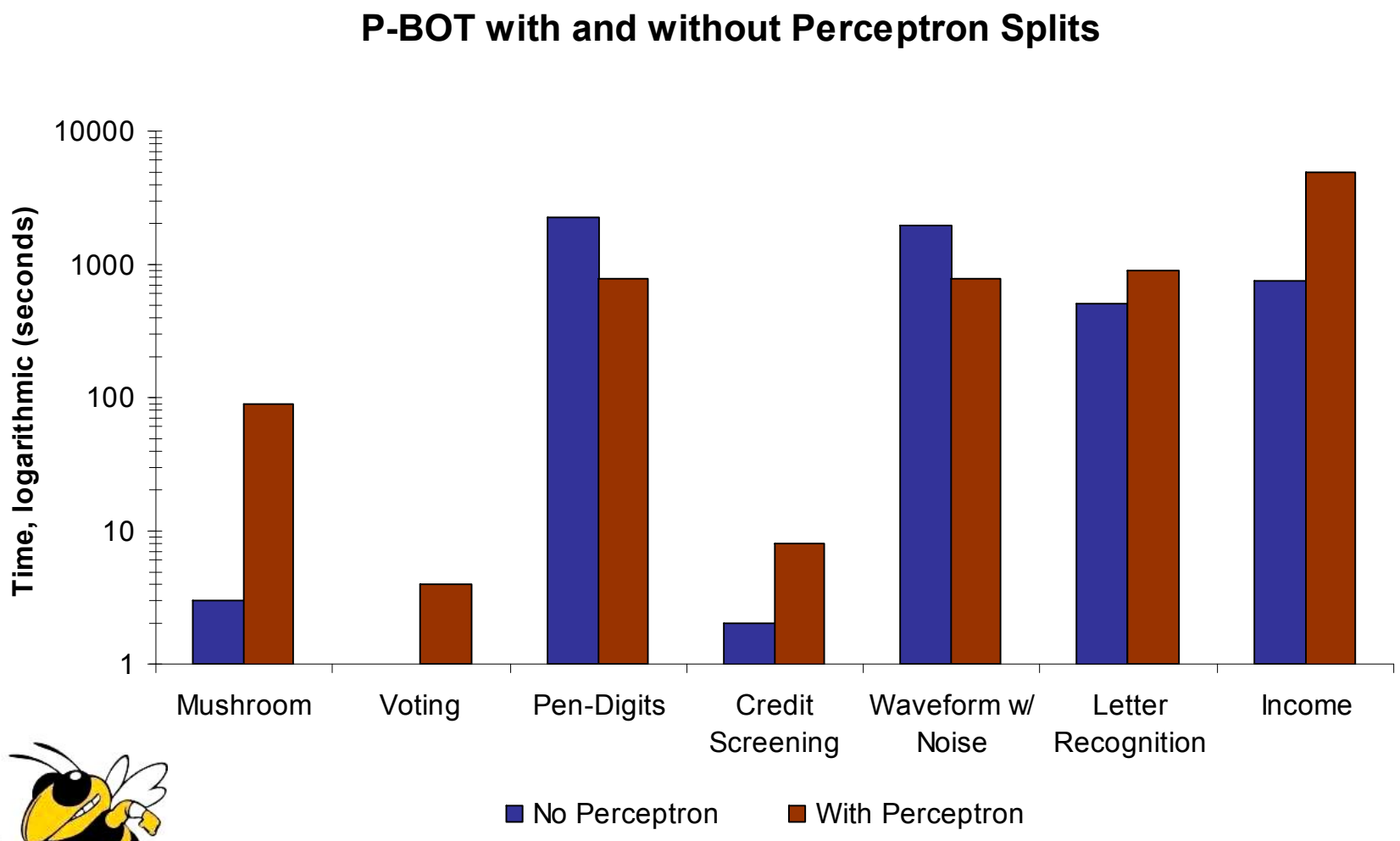
Results


Total Wall Clock Time for Various Datasets
Time (sec)
6000
5000
4000
3000
2000
1000
0
Mushroom
Voting
Pen-Digits
Credit Screening
Waveform w/ Noise
Letter Recognition
Income
J48
ANN
P-BOT

Results


Tree Size for Various Datasets
Size of Tree, Logarithmic (# of nodes)
1000
100
10
1
Mushroom
Voting
Pen-Digits
Credit Screening
Waveform w/ Noise
Letter Recognition
Income
J48
P-BOT

Results


P-BOT with and without Perceptron Splits
Time, logarithmic (seconds)
10000
1000
100
10
1
Mushroom
Voting
Pen-Digits
Credit Screening
Waveform w/ Noise
Letter Recognition
Income
No Perceptron
With Perceptron

Conclusions

- P-BOT combines the advantages of Decision Trees & Neural Nets
- Always faster than a NN with one hidden layer
- Consistently more accurate than J48
- More accurate than both on a variety of datasets

Future Work

- Code optimization for continuous attributes
- Empirical comparison of P-BOT to OC1 and LMDT algorithms
- Exploration of the optimal number of epochs

References

- S. Murthy, S. Kasif, S. Salzberg, and R. Beigel. (1993) *OC1: Randomized induction of oblique decision trees.* In Proceedings of the Eleventh National Conference on Artificial Intelligence, pages 322--327, Boston, MA, 1993. MIT Press.
- Utgoff, P. E., & Brodley, C. E. (1991). *Linear machine decision trees*. Tech. rep. 10, University of Massachusetts at Amherst.
- Gideon Pertzov. http://gpdev.net/NeuroDriver_bpnet.html
- Mitchell, T. M. (1997) *Machine Learning*. McGraw-Hill
- http://mathworld.wolfram.com/Polytope.html