Report about the Tree Pruning Programs
Xiaoming Huo and Xuelei Ni
2004

Abstract
This document describes the software that was used in creating a classifier for identifying sparse
events in a sequence via applying tree-pruning techniques.

1 Main programs
File structure.m in the package describe the architecture of the MATLAB functions that are included.
If function A calls subfunction B, it will be denoted as ‘A ← B’.
The following are main functions.

• TreeBuild_4.m – given a sequence, build the original tree.
  Subfunctions: TreeBuild_4.m ← TreeToMatrix.m ← count.m.

• TreePurne_4.m – prune the original tree.
  Subfunctions: GetSubTree.m, GetNodeString.m, StringToNode.m and DeleteRow.m.

• PlotBinaryTree.m – given a binary tree having format as the tree generated by TreeBuild_4.m,
  this function can plot the tree and show the symbol associated with each branch and the number
  of subsequences associated with each node.
  Subfunctions: PlotBinaryTree.m ← FullBinaryTree.m ← FullLevel_2.m.

2 Demos
We give some figures, which illustrate the results of the tree pruning program.

2.1 Notations

• L: The maximal length of substrings under consideration. \( L + 1 \) is the depth of the unpruned
tree (if we consider the depth of the root node as 1).

• X: Sequence used to build the tree.

2.2 Results
Sections 2.2.1 and 2.2.2 give illustrations for deterministic sequences. Sections 2.2.3 and 2.2.4 are for
random sequences.
2.2.1 $X=1000011011$

Code: work01.m
Figure: Xsg1.eps and Xsg2.eps

We take $L = 4$. Figure 1 (a) is the unpruned 5-depth tree built from $X$ while Figure 1 (b) is the pruned tree.

![Figure 1: Unpruned (a) and pruned (b) tree from 1000011011.](image)

2.2.2 $X=0001000000$

Code: work02.m
Figure: XXsg1.eps and XXsg2.eps

This one is an extreme case. We only have one “1” in the whole sequence. We still take $L = 4$. The original and pruned trees are given in Figure 2 (a) and (b) respectively.

![Figure 2: Unpruned tree (a) and pruned tree (b) from 0001000000.](image)
2.2.3 \( X_i \) are i.i.d. Bernoulli(0.3)

L=4
Code: work03.m
Figure: IIDsg1.eps and IIDsg2.eps

L=6
Code: work04.m
Figure: IIDsg3.eps and IIDsg4.eps

In this case, we generate a binary sequence with length 500. I.e., \( X_i \)'s are independent and identically distributed random variables, which are from distribution Bernoulli(0.3). For \( L = 4 \), we give the original tree (Figure 3 (a)) and the pruned tree (Figure 3 (b)) for one realization of such sequence. We also give illustrations (Figure 4 (a) and (b)) for \( L = 6 \) for another realization of the sequence. For the case \( L = 6 \), the average depths of original tree and pruned tree are 7 and 3.5354, respectively.

![Figure 3: Unpruned (a) and pruned (b) tree from i.i.d Bern(0.3) sequence, with length 500.](image-url)
Figure 4: Unpruned tree (a) and pruned tree (b) from i.i.d Bern(0.3) sequence, with length 500. Here we enlarge $L$ to 6.
2.2.4 $X$ from a stochastic distribution

**Code:**
GenMarkov2.m – is used to generate the sequence;
DemoTreeProgram.m – is used to give the simulation result for one realization of this stochastic sequence;
**Figure:** Markovsg1.eps and Markovsg2.eps

The length of the sequence in this case is still 500. This is also a binary sequence. The sequence satisfies the following stochastic distribution:

$$X_i = \begin{cases} 
1, & \text{if } X_{i-1} = 0, \text{and } X_{i-2} = 0; \\
0, & \text{if } X_{i-1} = 1, \text{and } X_{i-2} = 1; \\
1 \text{ with probability } p_1, & \text{if } X_{i-1} = 1, \text{and } X_{i-2} = 0; \\
1 \text{ with probability } p_2, & \text{if } X_{i-1} = 0, \text{and } X_{i-2} = 1; 
\end{cases}$$

where $p_1$ and $p_2$ are two prescribed probabilities.

Figure 5 (a) is a complete tree with depth 6. Figure 5 (b) is the pruned version of the same tree. We can see that lots of nodes have been pruned.

Figure 5: An unpruned tree (a) and pruned tree (b) from a Markovian Sequence.
We can set \( p_1 \) and \( p_2 \) as random variables from Uniform(0,1) distribution and do the simulation for 1000 times. The following figure is the histogram of the average depth of the trees after pruning while \( L = 6 \) (depth of unpruned tree is 7).

Code:
- SimuMarcov.m is used to give the histogram;
- AvgDepth.m is used to calculate the average depth of a tree.

Figure: Markovsg3.eps

3 Conclusion

This document describes the software that we wrote during a project to study tree pruning technology in building an efficient classifier to locate sparse events in an extreme long sequence. We will continue with more research when time permits.