Lecture 6.44

Applications of Binary Tree (Continued)

Agenda

- Review of Binary Tree Applications
- Deep Dive into Expression Trees
- Evaluating Expressions using Trees
- Understanding Huffman Coding Trees
- Introduction to Decision Trees
- Summary and Q&A

What is an Expression Tree?

- A special kind of binary tree used to represent mathematical expressions.
- Internal Nodes: Store operators (+, -, *, /).
- Leaf Nodes: Store operands (variables or constants).
- Enables unambiguous evaluation of expressions without parentheses.
- Can represent postfix, prefix, and infix expressions through tree traversals.

Structure of an Expression Tree

- Expression: (A + B) * (C - D)
- Root Node: '*'
- Left Subtree Root: '+' (Leaves: A, B)
- Right Subtree Root: '-' (Leaves: C, D)
- Inorder Traversal: A + B * C - D (Infix)
- Postorder Traversal: A B + C D - * (Postfix)

Evaluating an Expression Tree

- Algorithm for evaluation:
- 1. If node is a leaf, return its value.
- 2. Recursively evaluate left subtree.
- 3. Recursively evaluate right subtree.
- 4. Apply the operator at the node to the left and right results.
- Example Evaluation: ((3 + 4) * (5 - 2)) evaluates to 21.

Introduction to Huffman Coding

- A lossless data compression algorithm.
- Assigns variable-length codes to characters based on their frequency.
- More frequent characters get shorter codes.
- Uses a binary tree called a Huffman Tree to generate prefix-free codes.
- Extensively used in file compression formats (e.g., ZIP, JPEG).

Building the Huffman Tree

- 1. Calculate frequencies of all characters.
- 2. Create leaf nodes for each and add to a priority queue.
- 3. While more than one node in queue:
- a. Remove two nodes with lowest frequency.
- b. Create a new internal node with sum of their frequencies.
- c. Add the new node back to queue.
- 4. The remaining node is the root.

Huffman Coding in Action

- Characters: A(5), B(2), C(1), D(1)
- Merge C(1) & D(1) -¿ CD(2)
- Merge B(2) & CD(2) -¿ BCD(4)
- Merge A(5) & BCD(4) -¿ Root(9)
- Codes generated by traversing from root:
- Left edge = 0, Right edge = 1
- A = 0, B = 10, C = 110, D = 111

Decision Trees

- A predictive modeling tool in Machine Learning and AI.
- Internal Nodes: Represent a 'test' on an attribute.
- Branches: Represent the outcome of the test.
- Leaf Nodes: Represent class labels or decisions.
- Used for classification and regression tasks.

Structure of a Decision Tree

- Root Node: Best predictor attribute.
- Splitting: Dividing nodes based on condition.
- Pruning: Removing sections to prevent overfitting.
- Example Question: Is it raining?
- Yes -¿ Go Left (Stay inside)
- No -¿ Go Right (Go outside)

Summary

- Expression Trees represent and evaluate mathematical equations.
- Huffman Trees optimize data compression via prefix-free codes.
- Decision Trees facilitate complex logical and predictive decisions.
- Binary trees are versatile structures serving fundamentally different computational needs.

Next Lecture Preview

- Recap of the entire Binary Trees unit.
- Complex problem-solving using trees.
- Practice problems for Binary Search Trees.
- Preparing for practical implementations in Python.