Data Structures
Reminders

• Homework B: Complexity & Recursion
  – Out: Thu, Nov. 8
  – Due: Tue, Nov. 20 at 11:59pm

• Quiz A: Logic & Proofs; Computation
  – Mon, Nov. 26, in-class
  – Covers Lectures 1 – 6
Q&A
DYNAMIC PROGRAMMING
Hidden Markov Model

A Hidden Markov Model (HMM) provides a joint distribution over the tunnel states / travel times with an assumption of dependence between adjacent tunnel states.

\[ p(O, S, S, O, C, 2m, 3m, 18m, 9m, 27m) = (0.8 \times 0.08 \times 0.2 \times 0.7 \times 0.03 \times \ldots) \]
Forward-Backward Algorithm: Finds Marginals

Motivating Example

\[ p(v \ a \ n) = \left( \frac{1}{Z} \right) \times \text{product weight of one path} \]

- Marginal probability \( p(Y_2 = a) \)
- \( Z \) is the total weight of all paths through \( a \)
Constituency Parsing

• **Variables:**
  - Constituent type (or $\emptyset$) for each of $O(n^2)$ substrings

• **Interactions:**
  - Constituents must describe a binary tree
  - Tag bigrams
  - Nonterminal triples (parent, left-child, right-child)

[these factors not shown]

Motivating Example

(Naradowsky, Vieira, & Smith, 2012)
Dynamic Programming

**Key Idea:** Divide a large problem into **reusable** subproblems and solve each subproblem, storing the result of each for later reuse

“Let’s take a word that has an absolutely precise meaning, namely dynamic, in the classical physical sense. It also has a very interesting property as an adjective, and that is it’s impossible to use the word, dynamic, in a pejorative sense. Try thinking of some combination that will possibly give it a pejorative meaning. [...] Thus, I thought **dynamic programming** was a good name. It was something not even a Congressman could object to. So I used it as an umbrella for my activities.”

Dynamic Programming

Chalkboard:
– Big Idea: Dynamic Programming
– Example: Fibonacci with and without dynamic programming
  • Recursive Fibonacci’s computational complexity
  • Dynamic programming Fibonacci’s computational complexity
– Types of Dynamic Programming
  • Tabulation (bottom-up)
  • Memoization (top-down)
– Example: Matrix Product Parenthesization
DATA STRUCTURES FOR ML
# Abstractions vs. Data Structures

<table>
<thead>
<tr>
<th>Abstractions</th>
<th>Data Structures</th>
</tr>
</thead>
<tbody>
<tr>
<td>• List</td>
<td>• Array (fixed size)</td>
</tr>
<tr>
<td>• Set</td>
<td>• Array (variable size)</td>
</tr>
<tr>
<td>• Map</td>
<td>• Linked List</td>
</tr>
<tr>
<td>• Queue (FIFO)</td>
<td>• Doubly-Linked List</td>
</tr>
<tr>
<td>• Stack (LIFO)</td>
<td>• Multidimensional Array</td>
</tr>
<tr>
<td>• Graph</td>
<td>• Tensor</td>
</tr>
<tr>
<td>• Priority Queue</td>
<td>• Hash Map</td>
</tr>
<tr>
<td></td>
<td>• Binary Search Tree</td>
</tr>
<tr>
<td></td>
<td>• Balanced Tree</td>
</tr>
<tr>
<td></td>
<td>• Trie</td>
</tr>
<tr>
<td></td>
<td>• Stack</td>
</tr>
<tr>
<td></td>
<td>• Heap</td>
</tr>
<tr>
<td></td>
<td>• Graph</td>
</tr>
<tr>
<td></td>
<td>• Bipartite Graph</td>
</tr>
<tr>
<td></td>
<td>• Sparse Vector</td>
</tr>
<tr>
<td></td>
<td>• Sparse Matrix</td>
</tr>
</tbody>
</table>
Data Structures for ML

Examples...

• Data:
  – Dense feature vector (array)
  – Sparse feature vector (sparse vector)
  – Design matrix (multidimensional array)

• Models:
  – Decision Trees (tree)
  – Bayesian Network (directed acyclic graph)
  – Factor Graph (bipartite graph)

• Algorithms:
  – Greedy Search (weighted graph)
  – A* Search (priority queue/heap)
  – Forward-backward for HMM (trellis)
Trees

**Chalkboard:**

– Binary Tree
  - Representation
  - Depth First Search
    – pre-order traversal
    – in-order traversal
    – post-order traversal
  - Breadth First Search

– Decision Tree
  - Representation
Tree Traversals

- Pre-order
- In-order
- Post-order

Depth First Search

Breadth First Search

Figures from Wikipedia
Graphs

Chalkboard:

– Undirected Graphs
  • Representation
  • Breadth First Search
In-Class Exercise
• Suppose we now have an undirected (possibly cyclic) graph
• You need a breadth-first traversal of the graph from some query node
• Your friend suggests you use the same BFS algorithm we defined for binary trees

1. What goes wrong?
2. How can you fix it?
Sparse Vectors

*Chalkboard:*

- Sparse Vector
  - Representation
  - Sparse Dot Product
  - Addition of dense vector and sparse vector
Data Structures & Algorithms

Chalkboard:
– Weighted Directed Acyclic Graph
  • Representation
  • Greedy Search
  • Dijkstra’s Algorithm
  • A* Search
– Binary Search Tree
  • Representation
  • Average vs. Worst Case Time Complexity
  • Search
  • Insertion
  • Deletion