Introduction to: Computers & Programming: Sample Programs with Conditionals

Adam Meyers
New York University
Examples of Decision Tree Programs

• Automatic Bank Teller Machines
• Expert Systems
• Automated Phone Systems
• Interactive Fiction and Similar Games
  – Educational software and Children's Stories
  – Adventure-type games
Example 1: Bank Teller Machine

• Flow Charts
    • Page 9

• Flow Chart symbols: conventions seem to vary; some additional shapes
  – Circle (or Ovals) = Start/End/Continue
  – Rectangle (or Ovals) = Commands
  – Diamonds (or Vertical Bars) = Decisions
  – Parallelograms (slanted rectangles) = Input/Output
Example 2: A (toy) Expert System to Distinguish a Cold from the Flu

- **Source:**

- **1\(^{st}\) Step:** Sum up all the factors involved

- **2\(^{nd}\) Step:** Model them as a decision tree, an organized series of yes/no questions

- **3\(^{rd}\) Step:** Implement them as a Python program
<table>
<thead>
<tr>
<th>Symptom</th>
<th>Cold</th>
<th>Flu</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fever</td>
<td>Mild, more common in children</td>
<td>Usually Higher (100° to 102° F), lasts 3-4 days</td>
</tr>
<tr>
<td>Headache</td>
<td>Occasional</td>
<td>Common</td>
</tr>
<tr>
<td>Aches/Pains</td>
<td>Slight (implied not always)</td>
<td>Usual, often severe</td>
</tr>
<tr>
<td>Fatigue, weakness</td>
<td>Sometimes</td>
<td>Usual, can last 2 to 3 weeks</td>
</tr>
<tr>
<td>Extreme Exhaustion</td>
<td>Never</td>
<td>Usual, at beginning of illness</td>
</tr>
<tr>
<td>Stuffy Nose</td>
<td>Common</td>
<td>Sometimes</td>
</tr>
<tr>
<td>Sneezing</td>
<td>Usual</td>
<td>Sometimes</td>
</tr>
<tr>
<td>Sore Throat</td>
<td>Common</td>
<td>Sometimes</td>
</tr>
<tr>
<td>Chest Discomfort, Cough</td>
<td>Mild to moderate; hacking cough (implied not always)</td>
<td>Common; can become severe</td>
</tr>
</tbody>
</table>
Info about Colds Taken from text

- **Duration:**
  - Contagious for a few days
  - Symptoms last about one week
    - If more than a week, may be bacterial infection or allergic rhinitis (hay fever) – allergic reaction

- **Symptom1:** sore throat for 1-2 days
- **Symptom2:** runny nose, congestion
- **Symptom3:** cough (after 4-5 days)
- **Variable symptom:**
  - fever in children
  - possibly slight fever in adults

- **Caused by several hundred different viruses**
- **Complications:** sinus congestion; middle ear infection
Info about the Flu taken from text

- Two types of Flu: Seasonal and Swine Flu
- Symptoms: sore throat, fever, headache, muscle ache, soreness, congestion, cough
- Swine flu specific symptoms: vomiting and diarrhea
- Duration: a few days to a few weeks
- Symptoms can take a few days to a week to dissipate
- Possible complications
  - Pneumonia (possibly life threatening)
    - Symptom: shortness of breath
    - A fever that goes away and then returns after 1-2 days
  - Others: sinusitis, bronchitis, ear infection
How can we model the Information in the article in our program?

• The WebMD article is not written in formal language – we have to interpret it so we can codify it in a program

• Many symptoms indicate either cold or flu
  – Some of these have informal frequencies associated with cold and flu as indicated by words like:
    • never, occasional/sometimes/Mild/Slight, common, usual
  – We can interpret these using a point system that we share with the user, e.g.,
    • never = 0, occasional/sometimes/mild/slight = 1, common = 3, usual = 6, always = 10
  – Severity of Symptoms can be treated the same way:
    • Nonexistant = 0, mild = 1, moderate = 5, extreme = 10
More Considerations for Modeling the Problem

• The text provides clues that are not in the table
• It mentions which symptoms occur first
• It tells about symptoms specific to Swine Flu, a subtype of Flu
• Some of the questions imply human knowledge that we have to incorporate into a program.
  – A fever is a temperature that is probably at least 99°F
  – A child is probably someone who is under a certain age
• We are guessing that age is 16 for purposes of this program
Generalizations and Simplifications

- Assumption: These are common symptoms of many ailments. We need at least 3 symptoms before guessing that Flu or Cold is a possible diagnosis. This will prevent false diagnoses.

- We can divide all the symptoms into the following classes:
  - Symptoms that absolutely favor Flu over Cold
  - Symptoms that tend to favor Flu
  - Symptoms that tend to favor Cold
  - Symptoms that absolutely favor Swine Flu over Seasonal Flu

- We can try a voting scheme
  - We don't know if this will work, but we can test it
  - This is for demo purposes only. We won't have an extensive testing phase and must assume the program is not accurate.
Assumptions in Our Program

• Our list of symptoms:
  – fever, tiredness, headache, fatigue, other aches and pains, chest discomfort and coughing, stuffy/runny nose, sneezing, sore throat
  – These can be true/false or have a range of values

• Symptoms absolutely favoring Flu
  – High level of fatigue or high temperature

• Symptoms favoring cold
  – low fever in children, sneezing, sore throat, stuffy/runny nose
More Assumptions in Our Program

• Symptoms favoring flu
  – headache, other aches, medium level of fatigue, coughing, illness longer than 1 week

• Symptoms absolutely favoring Swine over Seasonal Flu
  – vomiting and diarrhea

• Definition of Child for our purposes: Age < 16

• Fever: temperature >= 99
  – Low: temperature >= 99 and temperature <=100
  – High: temperature > 100
Our Algorithm
Ways We Could Improve the Program

• Include information about the sequence of symptoms.
  – Colds often begin with sore throats, which go away after a few days and are followed by nasal symptoms

• Consult other articles

• Consult a doctor

• Test the accuracy of the program on real data (real instances where we know the diagnosis and symptoms).
  – Modify the program to better account for the data
  – Test the program on new data
Implementation Details:
The function is_yes_or_no

- Takes one argument: the question to be asked
- Uses `input` function to ask the question and retrieve answer
- Converts yes or no answer into True or False
- Assumes unexpected answers are equivalent to 'No'
  - Alternative: Give user an error and exit the program
  - Alternative: Give user an error and ask for Yes or No again
- Uses Counters: symptoms, flu_symptoms, cold_symptoms
  - These are incremented by 1 when we identify a new symptom that fits the appropriate category
    - `counter = 1 + counter`
  - Some of the boolean tests involve counters and boolean operators (`>`, `<`, `>=`, `<=`, `==`, `/=`)
More Implementation Details

• \textit{number\_from\_zero\_to\_ten}
  – Makes sure that an integer from 0 to 10 is used
  – Rounds to the nearest integer
  – There would be an error if the user entered a non-number – there will be a section on proper error handling later this term.

• \textit{check\_for\_swine\_flu}
  – There are exactly 2 symptoms where either is evidence of Swine Flu, provided that flu is a possibility
  – One simple yes/no question covers this
  – This function is separated to make it easy to revise this in the future, should additional information be incorporated in the program.
Our System vs. Real Medical Expert Systems

• We need to include a warning in this program that the diagnosis should not be taken seriously

• Before releasing a real expert system, we would test it extensively and modify it so it performs accurately.
  – There was no quality control for this program

• Doctors are consulted for real systems – systems are not based on web articles written for non-doctors

• One expert system designed by a doctor is available online at: http://easydiagnosis.com/
  – Dr. Schueler, who designed this system, also warns that this program should not take the place of a real doctor
Expert Systems

• These can be represented by decision trees
  – They attempt to model human reasoning based on the order in which a human being would ask questions.

• Of course, there are other models for automatically making the same sorts of decisions
  – For example, predictions can be based on statistical correlations

• They are used in many fields: medicine, fixing machinery, how to choose a wine, picking an airplane flight, etc.

• Information on expert systems:
  http://edutechwiki.unige.ch/en/Expert_system
Sample Topics for Homework Problem

• Voting Advice:
  – Depends on: number of candidates, relative political affiliations of candidate and voter, relative political views of candidate and voter, perceived ability of candidate to win (and whether this matters to the voter), severity of misdeeds of candidate, importance of good works of candidate, candidate's previous experience, etc.

• Choosing Wine:
  – Depends on budget, what you are eating, what is available where you shop, personal preferences about sweetness, color, aroma, etc.

• Other: classification of biological specimen, predicting natural disaster (volcano, earthquake), determining the value of a used car, etc.
Summary

• Homework Assignment 5
  – An opportunity to think through a logical problem that you have a lot of knowledge about and structure the information as a program
  – Or write a type of fiction that includes lots of variables, but that could also be carefully thought out

• There has been a lot of previous work on both these sorts of programs.