DD2380 Artificial Intelligence
Homework 3 Review

October 7, 2010

a) Sounds about right. This is how you would generate both matrices. I'm unsure about the last line, estimating the model is the same as generating the new model, and it doesn't have to be compared, because you're estimating the model, not how accurate it is.

Result: *2 - Good* - Understands how to train both translation and observation matrices.

b) Good, correctly calculated probabilities. $P(o_2 = H_2 | x_2 = LEFT)$ taken correctly from the table.

Result: *2 - Good*

c) Doesn't calculate the correct sequence, compared using the Viterbi algorithm Python implementation on Wikipedia, and my own implementation which generates the same sequence, ['RIGHT', 'FORWARD', 'FORWARD', 'FORWARD', 'STOP', 'STOP', 'STOP', 'STOP', 'STOP', 'STOP']. I've gotten the same sequence from others.

The program, unfortunately crashes. I take this as last-second changes that accidentally got submitted.

Although the implementation doesn't seem to be exactly correct, s/he seems to have gotten the overall hang of the Viterbi algorithm, which I think deserves partial credit.

Results: *1 - OK* - Incorrect, but seems to understand the overall algorithm.

```
$ cc -g -o viterbi viterbi.c
$ ./viterbi
Segmentation fault
$ valgrind ./viterbi
==10412== Memcheck, a memory error detector.
==10412== Copyright (C) 2002-2008, and GNU GPL'd, by Julian Seward et al.
==10412== Using LibVEX rev 1884, a library for dynamic binary translation.
==10412== Copyright (C) 2004-2008, and GNU GPL'd, by OpenWorks LLP.
```

```
==10412== Using valgrind-3.4.1-Debian, a dynamic binary instrumentation framework.
==10412== Copyright (C) 2000-2008, and GNU GPL'd, by Julian Seward et al.
==10412== For more details, rerun with: -v
==10412==
==10412== Invalid write of size 4
==10412==    at 0x804867F: main (viterbi.c:74)
==10412==  Address 0x4 is not stack'd, malloc'd or (recently) free'd
==10412==
==10412== Process terminating with default action of signal 11 (SIGSEGV)
==10412==  Access not within mapped region at address 0x4
==10412==    at 0x804867F: main (viterbi.c:74)
==10412==  If you believe this happened as a result of a stack overflow in your
==10412==  program's main thread (unlikely but possible), you can try to increase
==10412==  the size of the main thread stack using the --main-stacksize= flag.
==10412==  The main thread stack size used in this run was 8388608.
==10412==
==10412== ERROR SUMMARY: 1 errors from 1 contexts (suppressed: 11 from 1)
==10412== malloc/free: in use at exit: 32 bytes in 2 blocks.
==10412== malloc/free: 2 allocs, 0 frees, 32 bytes allocated.
==10412== For counts of detected errors, rerun with: -v
==10412== searching for pointers to 2 not-freed blocks.
==10412== checked 52,104 bytes.
==10412==
==10412== LEAK SUMMARY:
==10412==    definitely lost: 0 bytes in 0 blocks.
==10412==      possibly lost: 0 bytes in 0 blocks.
==10412==    still reachable: 32 bytes in 2 blocks.
==10412==         suppressed: 0 bytes in 0 blocks.
==10412== Rerun with --leak-check=full to see details of leaked memory.
Segmentation fault
```

d) & e) Results: *0 - N/A* - Answers not available.