private question **2** views

"Vim Graph" and n-grams

To get the idea down correctly, we are going to simulate the user entering random key press, probably with some tool. I found a tool called xdotool that does this by reading a text file. Not really great. Then based on the probability of key presses, we could have some sort of markov chain. idk if I can get real probabilities from somewhere. Then show some distributions and explain stuff. Is there anything that I'm missing?

N-grams aren't so important right now, but I looked at this and it doesn't help me understand what's happening:
http://en.m.wikipedia.org/wiki/N-gram

If you know anything about this and can share, I'll appreciate it, but it's less than urgent. The Statistics project is way more important.

This private post is only visible to original poster and Instructors

Updated 1 day ago by Jean Pena

the students' answer, *where students collectively construct a single answer*

Click to start off the wiki answer

the instructors' answer, *where instructors collectively construct a single answer*

Hey Jean,

I fell asleep pretty early last night. Sorry I missed this. xdotool is the thing I was thinking of if you didn't want to use vimscript. And be very careful to have the focus be the vim window when you run the tool! It can go horribly wrong if the focus happens to be a shell @_@

To expand on the idea, I think there are a number of interesting questions you could ask. As parameters for the experiment, you could have

* input distribution of keystrokes
* starting state of document (from a fixed set, or maybe randomly generated from some distribution)

Now for possibly interesting questions:

* stationary distributions. I suspect that in general you will not have a stationary distribution, as the Markov chain will be transient, but maybe with certain settings of the parameters above, you will get interesting stationary distributions (I'm pretty certain I could come up with some contrived examples like this).
* statistics. Maybe looking at some skeletons might be interesting, for example, just looking at how the size of the document changes over time, or the number of words, etc.
* simulations. Surely you have seen these:

http://rubberducky.org/cgi-bin/chomsky.pl
http://thatsmathematics.com/mathgen/
http://pdos.csail.mit.edu/scigen/

It would be interesting to gather some data on frequencies (and maybe n-grams) for people typing C++, and then see what happens / whether or not it looks remotely like C++.

Anyway, I hope this helps. Sounds like it might be fun. Let me know how it turns out, or if you have any questions.

PS: regarding the n-grams, the basic idea is pretty simple. I have old books that perhaps show you where it first started, or you can just read wikipedia for the gist of it:
https://en.wikipedia.org/wiki/N-gram#n-gram_models

Updated 6 hours ago by wes

followup discussions *for lingering questions and comments*

(•) Resolved ( ) Unresolved

Jean Pena 22 hours ago I thought about it some more and came up with this probably more meaningful idea after discussing it with my group: As a goal we could show that an existing keyboard layout is very likely ideal for a programmer.

The steps, which range from trivial to "probably better to just copy and paste code" are somewhat like this:

1) ~~Use/write a keylogger(saw code for this somewhere).~~ vim -w does this. Have to find documentation on this.

2) Track keystrokes used when writing c++ code.(Project 7 is ideal for this. Each member could work on an assignment if they are working on some code, and we could merge the results for analysis).

3) Read the data and create a frequency table of keys pressed after a different key was pressed. Build transition probability matrix and compute it's nth power(something large to show some convergence).

4) idk what to do about distributions, but some nice graphs would be good to show. Maybe some percentile, etc stuff. At a minimum, we could use P^n to show that certain keys are very frequently pressed, in sequence or not, and having these keys on the home row would be the best and keyboard layout X does just that.

5) Then, if possible, using the original transition matrix, we could simulate the random keystrokes with these probabilties and probably output a text file that almost looks like C++.

Something along these lines.

Resolved Unresolved

Jean Pena 3 hours ago This is the first time I've seen those generators. I've certainly seen nonsense generators and spam emails. But the math and computer science research paper generators we pretty humorous.

Getting C++ code would be pretty amusing. For something like this to be sophisticated, it would be better to read a ton of source and make n-grams out of them. I still have to understand n-grams, but I can see some useful application. So that's a good start.

Besides that, I don't think we would be able to build something sophisticated enough to generate random code. But I started logging keystrokes today while I helped some Kingsborough students finish a project. It wasn't as successful as I had hoped. One of the students did the project from start to finish and the key logger stopped logging values after a few number pad strokes... I think I got half projects from 2 students.