David Zhao
Benjamin Quiring

November 23, 2018

1 Data

We downloaded a history of terminal commands for a UNIX computer from Perdue back in 1998. A sample of the data looked like:

```
cd
<1>
ll
vi
<1>
ll
emacs
```

Where a `<#>` represents the number of arguments given to a command. This was done to preserve the user's privacy. We cleaned the data as follows:

1. We combined a command with the number of arguments passed with it. For example `cd<1>` is different from `cd` which is also different from `cd<2>`.
2. We ignored operators like `; , -, |` and didn't consider them commands.
3. We ignored the `**SOF**` and `**EOF**` commands because they were added and not actually typed by the user.

After cleaning up the data as described above, we noticed that there were over 700 unique commands. We chose the 28 most common occuring commands to be the states in our Markov chain and proceeded to further filter the data by deleting commands that are not the 28 most common. The resulting dataset contained only the top 28. There were over twenty thousand time steps in the filtered data. We plotted the entire time series and the first 250 steps.

We calculated the occupation frequencies of the emperical data and graphed the distribution.

Even though the data is from 1998 the top 5 commands seem reasonable given our own experience using UNIX based operating systems. The majority of the time a user will be navigating their computer's file directory with commands like `cd` and `ll`

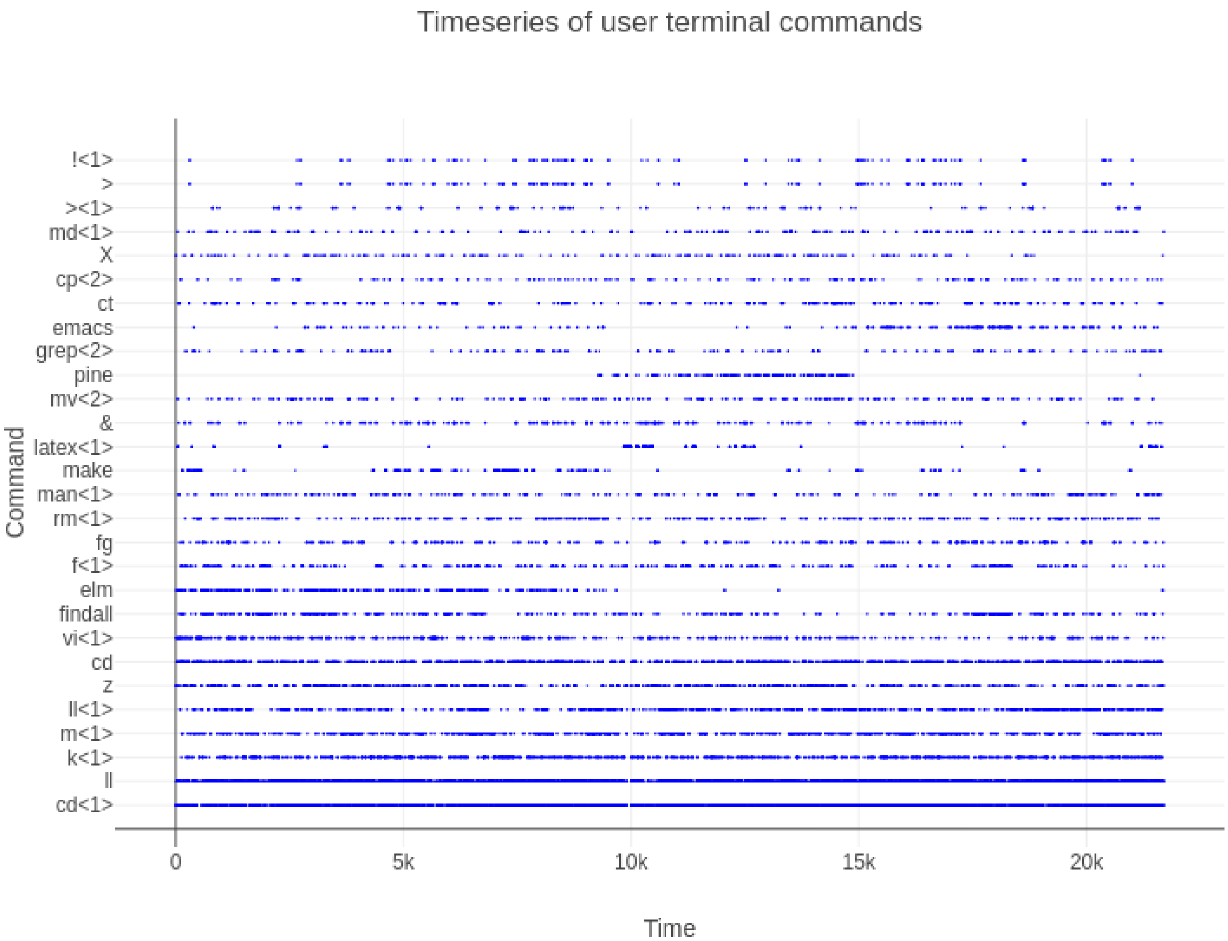
Next we computed the transition matrix by counting the number of times the chain went from state i to state j followed by dividing each count P_{ij} by the number of times we moved out of state i.

The transition matrix, plotted as a heatmap looked like:

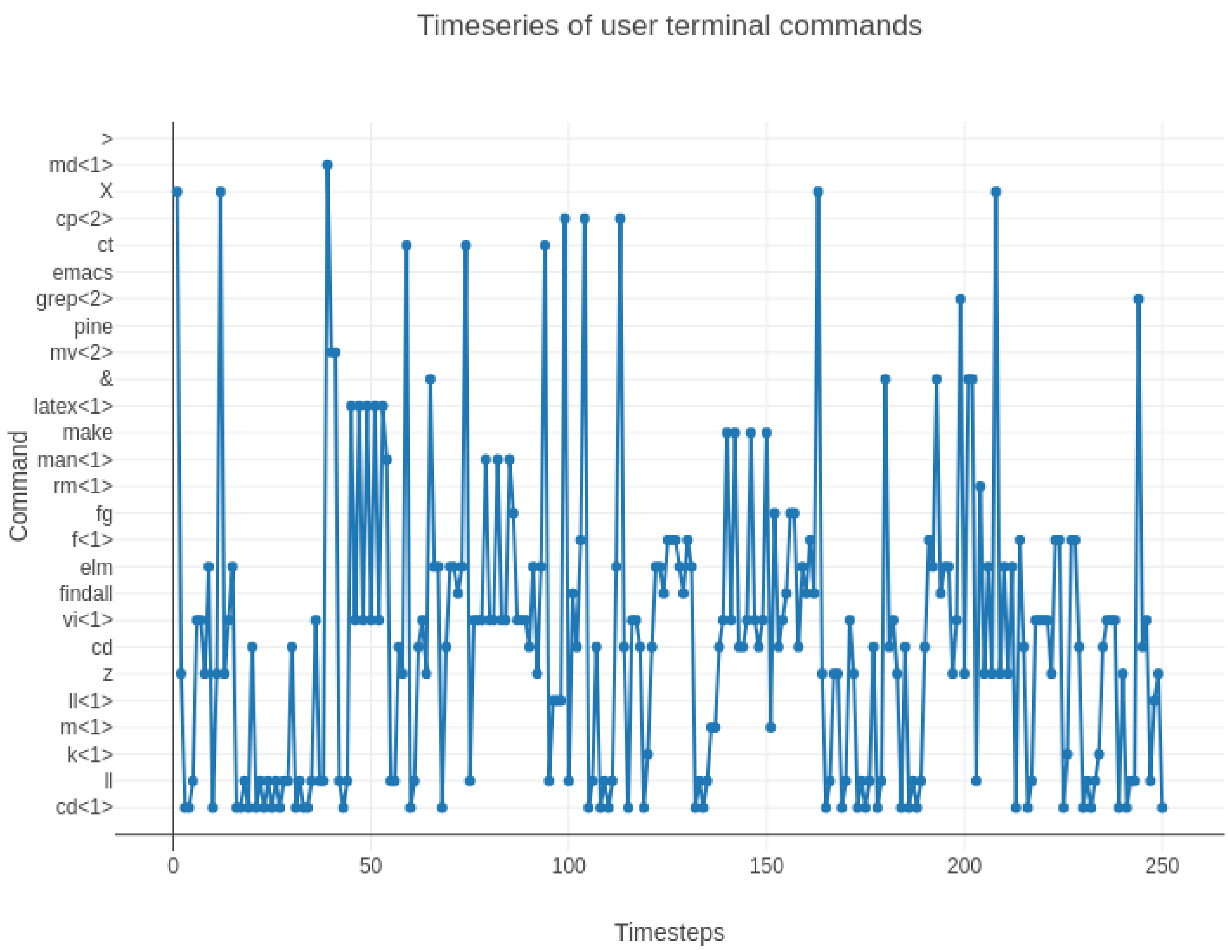
We raised the transition matrix to the power 1000 to compute the stationary distribution. We argue that the chain is irreducible because the data tracks the user's usage of the UNIX machine over two years. This means that there are many instances of logging in and out of the machine, which is done with the same set of commands every time (`rlogin` and `exit`). So the only instance where a state i is unreachable from j is if the user types a command and then doesn't proceed to logoff or doesn't proceed to log on. Both cases are impossible because a command isn't tracked until the user has logged on and since we're tracking the 28 most common commands there is no way that one of these commands are used so frequently while the user never exit afterwrods. Therefore, since the chain always visits the state `exit` every state can reach `exit` and as a result every state can reach every other state via the `exit` state.

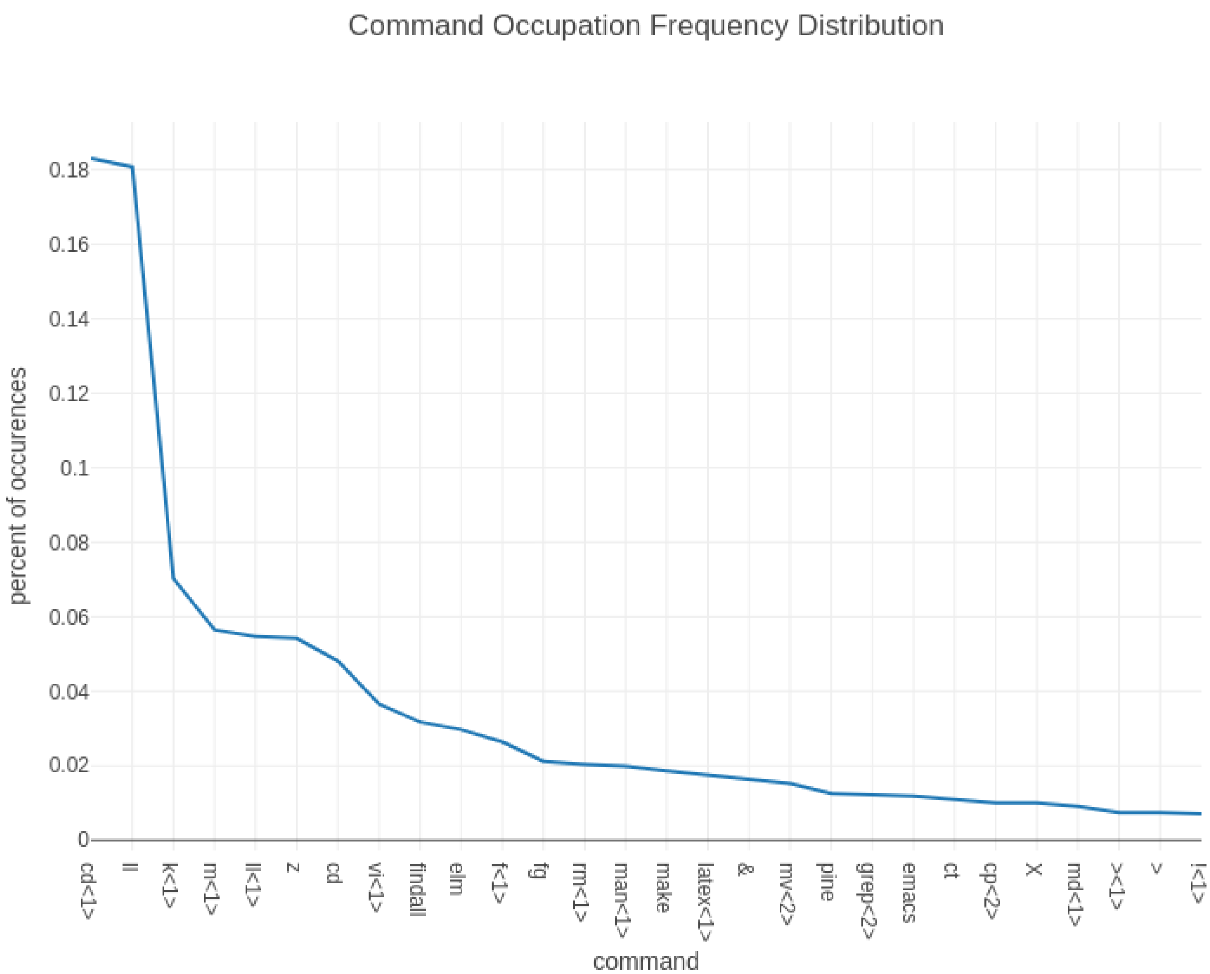
Therefore, when we plot the stationary distribution against the occupation frequency distribution, it is no surprise that the distributions are identical.

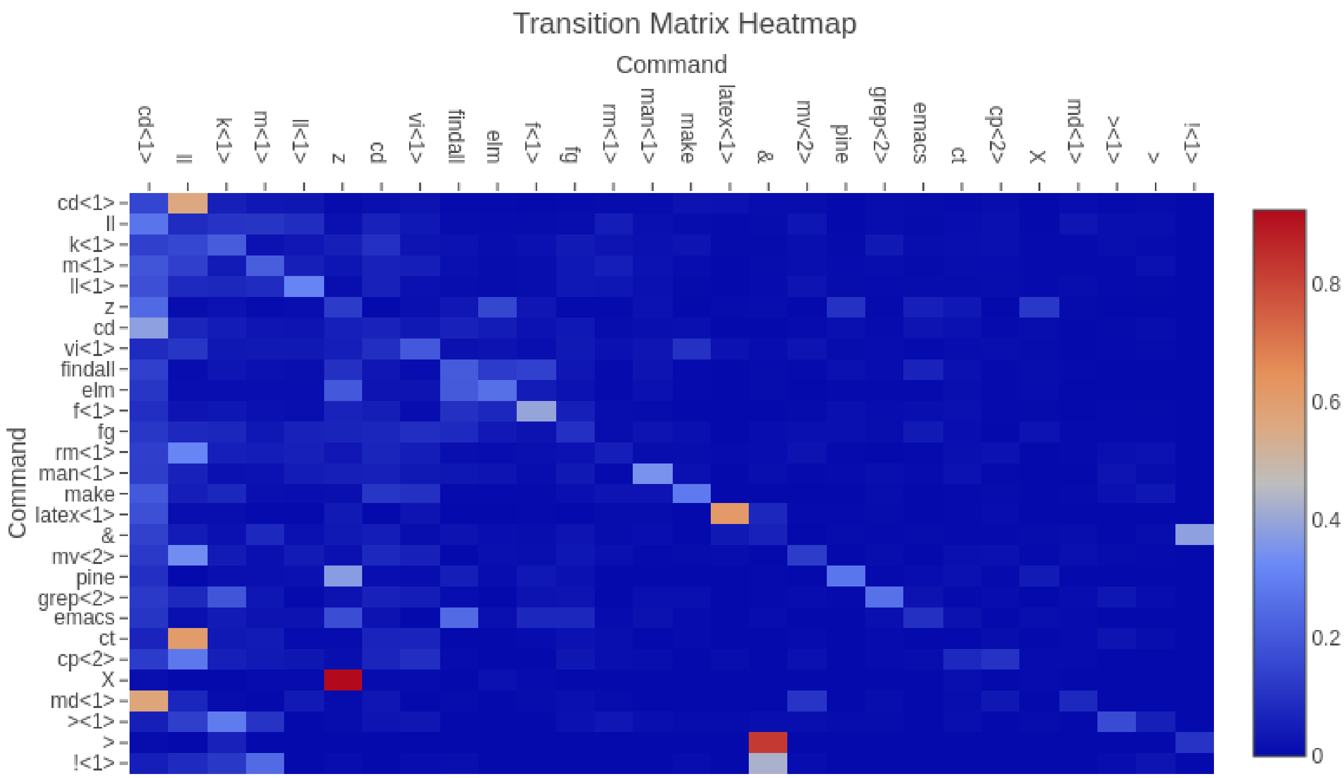
Finally we simluated the chain for 250 time steps. We assumed that the initial distribution was uniform over all the states.

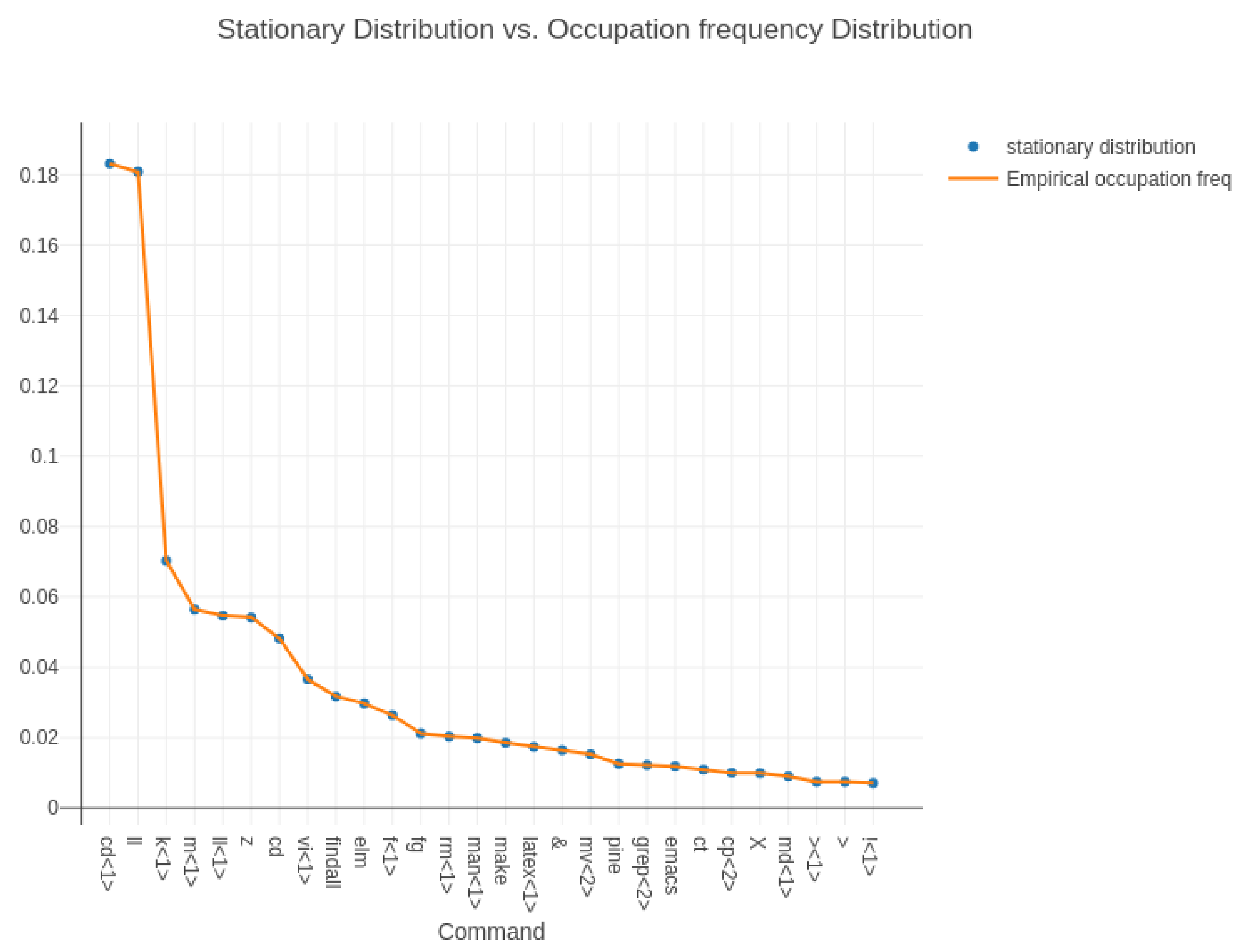


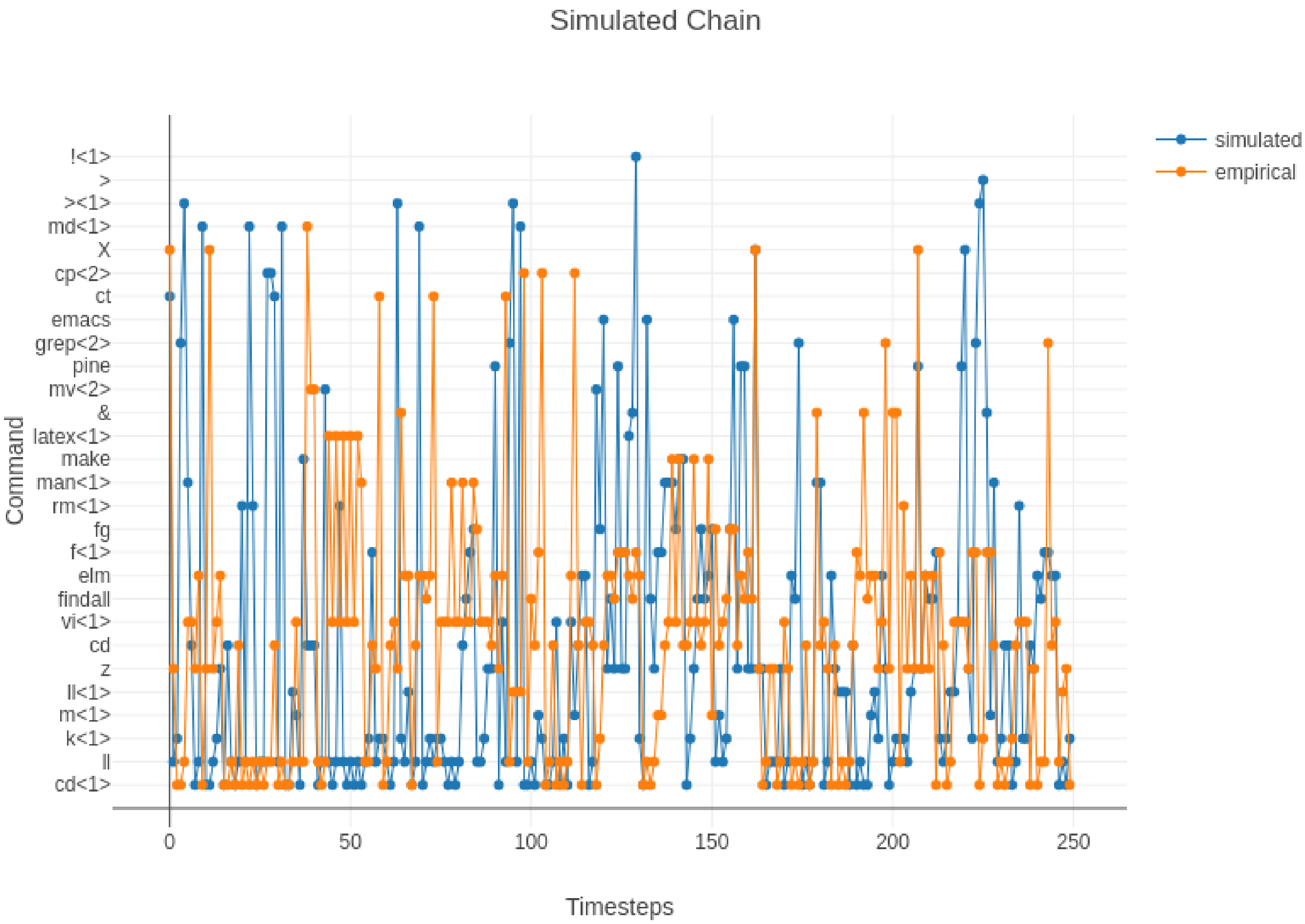
Finally we investigated the mixing time of the chain. We did this by, at each time step, computing the occupation frequency of the chain from the first to the last time step. At each step we calculate the occupational frequency of the states seen so far and subtracted this distribution with the calculated stationary distribution. We then plotted the l2 norm of the difference.

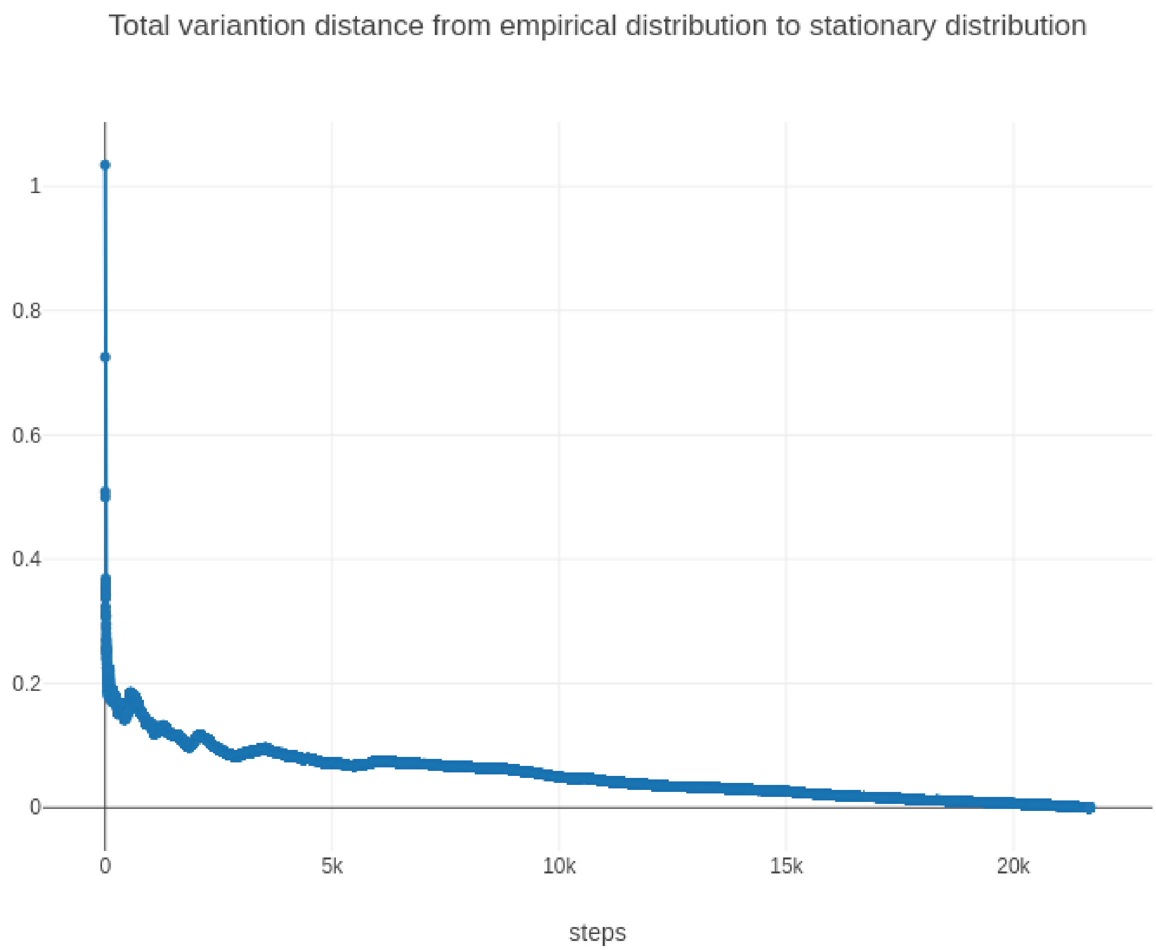

Timeseries of user terminal commands
>
md<1>
X
cp<2>
ct
emacs
grep<2>
pine
mv<2>
&
latex<1>
make
man<1>
rm<1>
fg
f<1>
elm
findall
vi<1>
cd
z
ll<1>
m<1>
k<1>
ll
cd<1>
Command
0
50
100
150
200
250
Timesteps


Command Occupation Frequency Distribution
percent of occurences
0.18
0.16
0.14
0.12
0.1
0.08
0.06
0.04
0.02
0
cd<1>
ll
k<1>
m<1>
ll<1>
z
cd
vi<1>
findall
elm
f<1>
fg
rm<1>
man<1>
make
latex<1>
&
mv<2>
pine
grep<2>
emacs
ct
cp<2>
X
md<1>
><1>
>
l<1>
command


Transition Matrix Heatmap
Command
cd<1>
ll
k<1>
m<1>
ll<1>
z
cd
vi<1>
findall
elm
f<1>
fg
rm<1>
man<1>
make
latex<1>
&
mv<2>
pine
grep<2>
emacs
ct
cp<2>
X
md<1>
><1>
>
!<1>
Command
0.8
0.6
0.4
0.2
0


Stationary Distribution vs. Occupation frequency Distribution
stationary distribution
Empirical occupation freq
0.18
0.16
0.14
0.12
0.1
0.08
0.06
0.04
0.02
0
cd<1>
ll
k<1>
rn<1>
ll<1>
z
cd
vi<1>
findall
elm
f<1>
fg
rm<1>
man<1>
make
latex<1>
&
mv<2>
pine
grep<2>
emacs
ct
cp<2>
X
md<1>
><1>
>
l<1>
Command


Simulated Chain
simulated
empirical
Command
Timesteps
!<1>
>
><1>
md<1>
X
cp<2>
ct
emacs
grep<2>
pine
mv<2>
&
latex<1>
make
man<1>
rm<1>
fg
f<1>
elm
findall
vi<1>
cd
z
ll<1>
m<1>
k<1>
ll
cd<1>
0
50
100
150
200
250


Total variantion distance from empirical distribution to stationary distribution
1
0.8
0.6
0.4
0.2
0
0
5k
10k
15k
20k
steps