

## AI ASSIGNMENT 2

Ganesh Prasanna Balakrishnan, Mohamad El-Rifai, Nishanth Shah, Rishi Khajuriwala

1) How did you select which node to update? Did you sweep through the nodes as a batch? Sample randomly? Something else? Justify your decision.

### ANSWER:

We swept through the nodes randomly for two reasons we thought were plausible:

- The number of iterations are around 10,000 or 100,000 and for the given size of tree (merely 8 nodes) the order of assignment of the nodes would not affect the performance of the algorithm because the query node would be visited almost as equally as other nodes.
- Assigning the nodes in this way also means that the update of the nodes takes place in a random way. There are algorithms where the nodes are updated in a particular sequence. This results in nodes being updated in the same sequence in every iteration. This would lead to a higher chance of the nodes being assigned to same value each time. Our process of random node selection for update eliminates this problem because of its stochastic and non-deterministic nature.

2) Experiment with using varying numbers of samples for each of the queries. How many samples are required for the estimated probability to converge? For each query plot the # of samples vs. estimated probability.

### ANSWER:

Query node - price

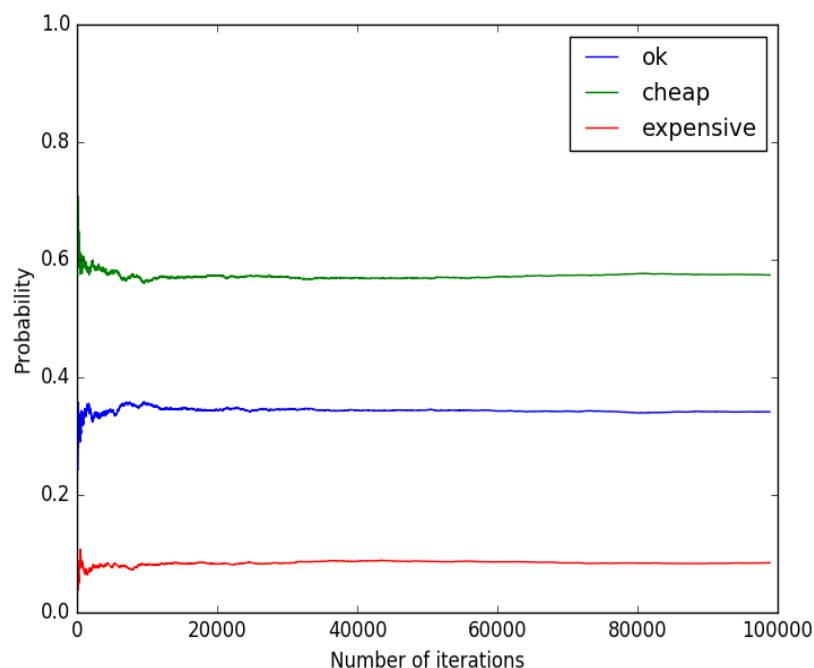
Evidence node - schools = good

location = ugly

Number of iterations = 100,000

Number of dropped samples = 1000

```
ganesh@gp:~/Ai$ python Gibbs.py price schools=good location=ugly -u 100000 -d 1000
P(price=cheap) = 0.574067366896
P(price=ok) = 0.341361825426
P(price=expensive) = 0.0845708076784
Time taken for 100000 iterations is 3.18978190422 s
```



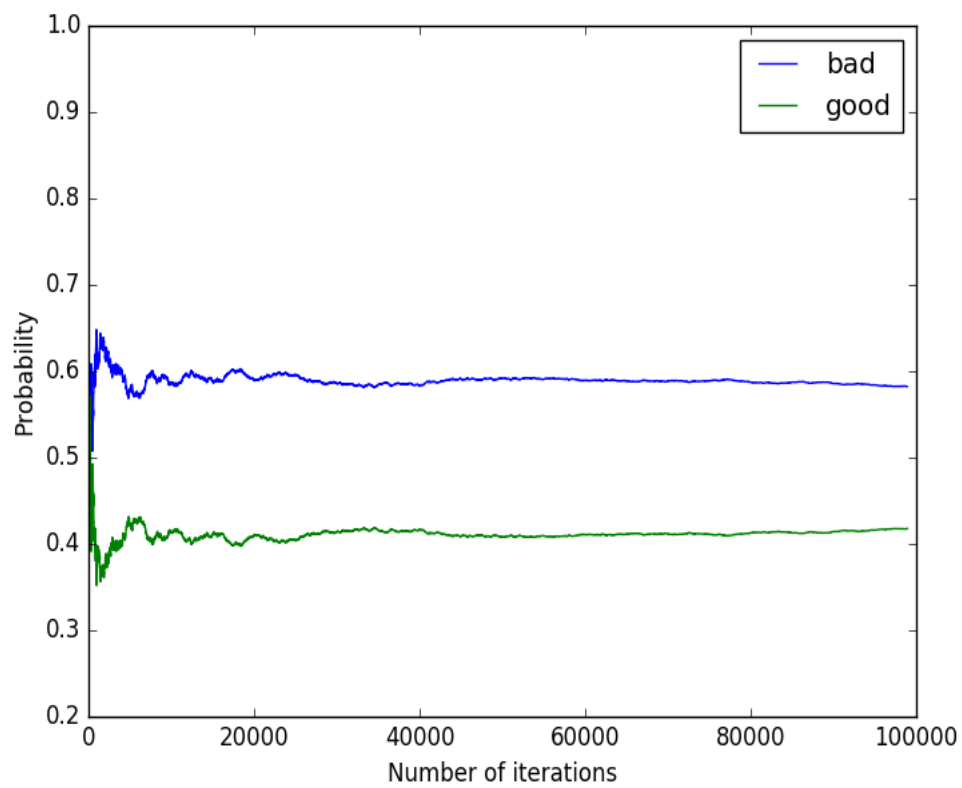
Query node - children

Evidence node - neighborhood = bad

Number of iterations = 100,000

Number of dropped samples = 1000

```
ganesh@gp:~/Ai$ python Gibbs.py children neighborhood=bad -u 100000 -d 1000
P(children=bad) = 0.5816448994
P(children=good) = 0.4183551006
Time taken for 100000 iterations is 3.05032515526 s
```



Query node - amenities

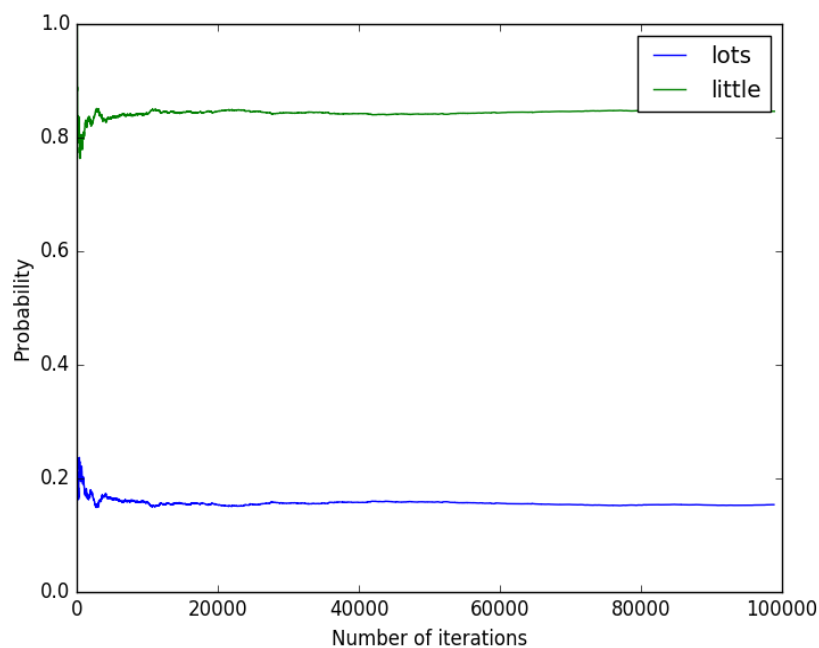
Evidence node - neighborhood = good

location = bad

Number of iterations = 100,000

Number of dropped samples = 1000

```
ganesh@gp:~/Ai$ python Gibbs.py amenities location=bad neighborhood=good -u 1000
00 -d 1000
P(amenities=lots) = 0.153738601824
P(amenities=little) = 0.846261398176
Time taken for 100000 iterations is 2.95061302185 s
```



The number of iterations it takes to converge for the cases mentioned above (discarding 1000 samples and running for 100,000 iterations) is around 5000.

### 3) What, if anything, seems to influence the number of samples required?

#### ANSWER:

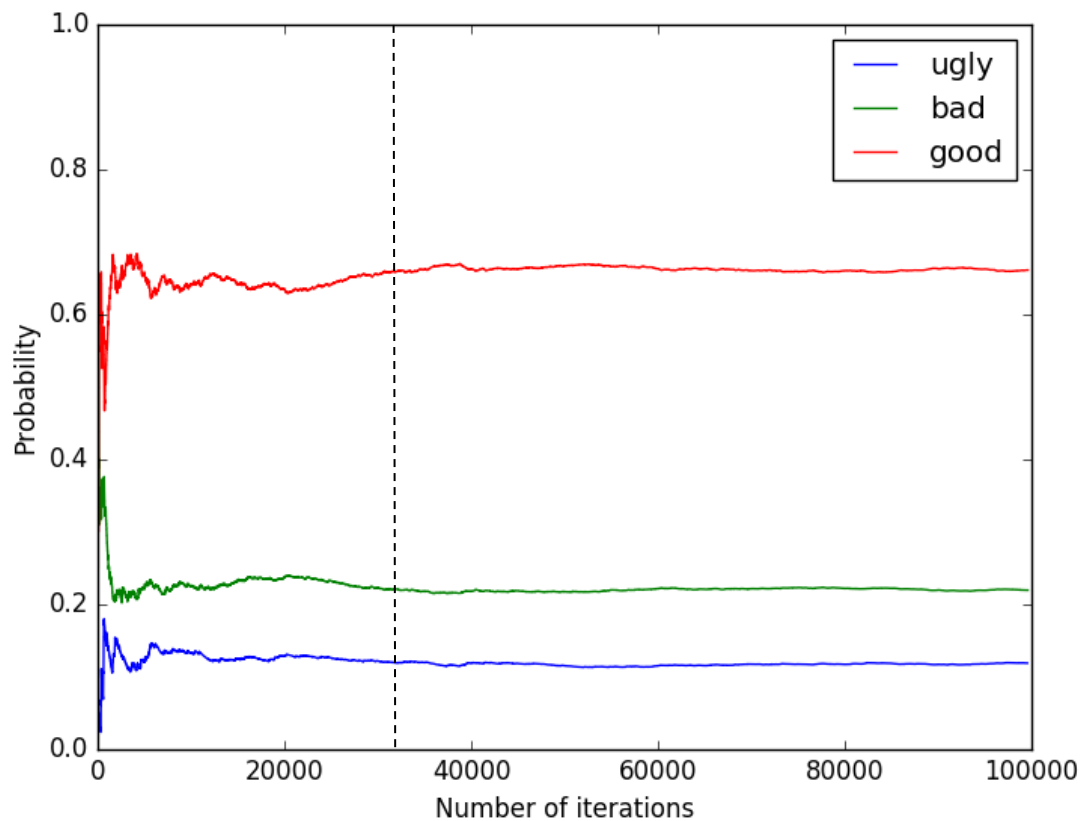
The number of evidence nodes influences the number of samples required. Higher the number of evidence nodes lower is the number of samples required.

#### 4) Does discarding initial samples speed the convergence of probabilities?

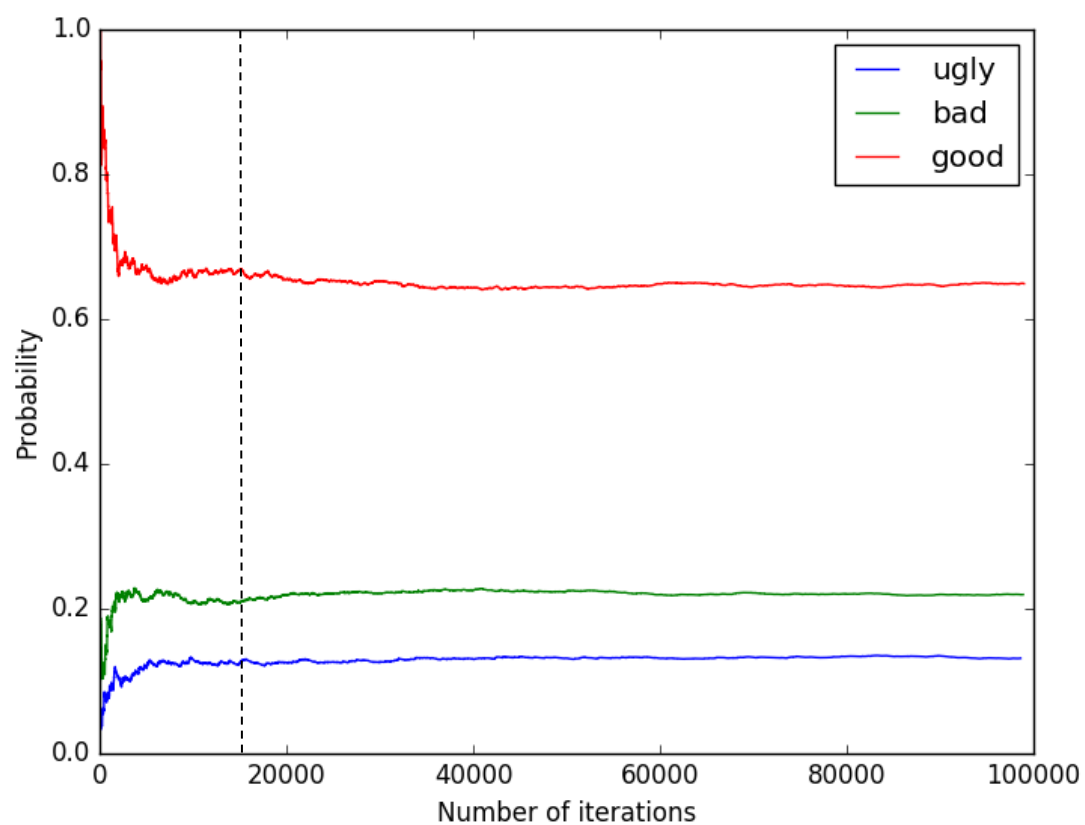
##### ANSWER:

Yeah discarding initial samples does increase the speed of convergence. We can see that clearly from the graphs below.

The dotted line represents the point at which the probabilities converge (approximately). We can see that the number of iterations for convergence of the probabilities with discarded samples is much lesser than the number of iterations taken for convergence without discarding any samples.



**100,000 iterations without discarded samples**



**100,000 iterations with 1000 discarded samples**