

AI ASSIGNMENT 2

Ganesh Prasanna Balakrishnan, Mohamad El-Rifai, Nishant 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.

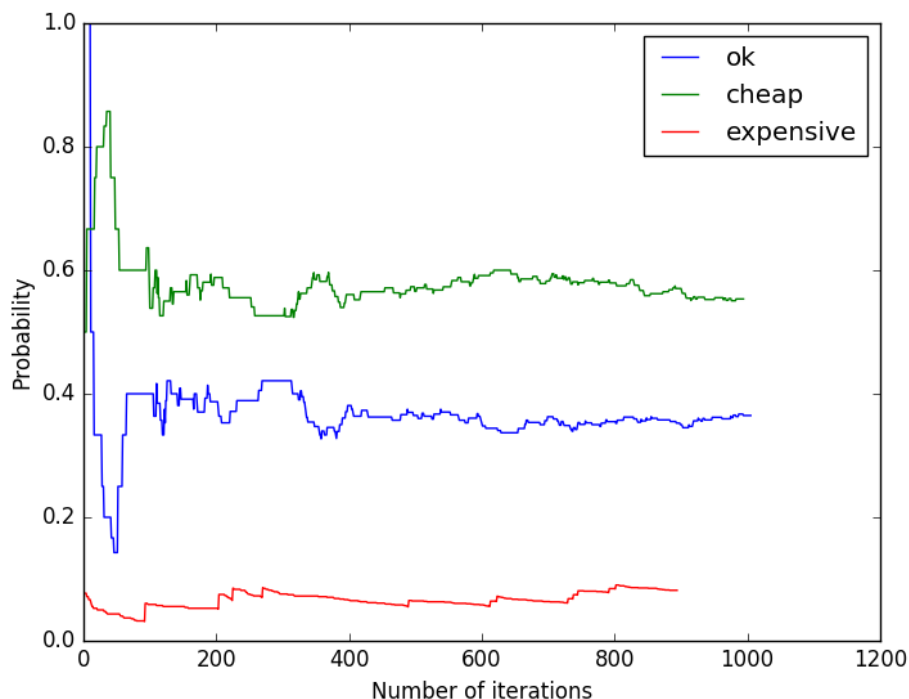
ANSWER:

a) Query node - price

Evidence node - schools = good
location = ugly

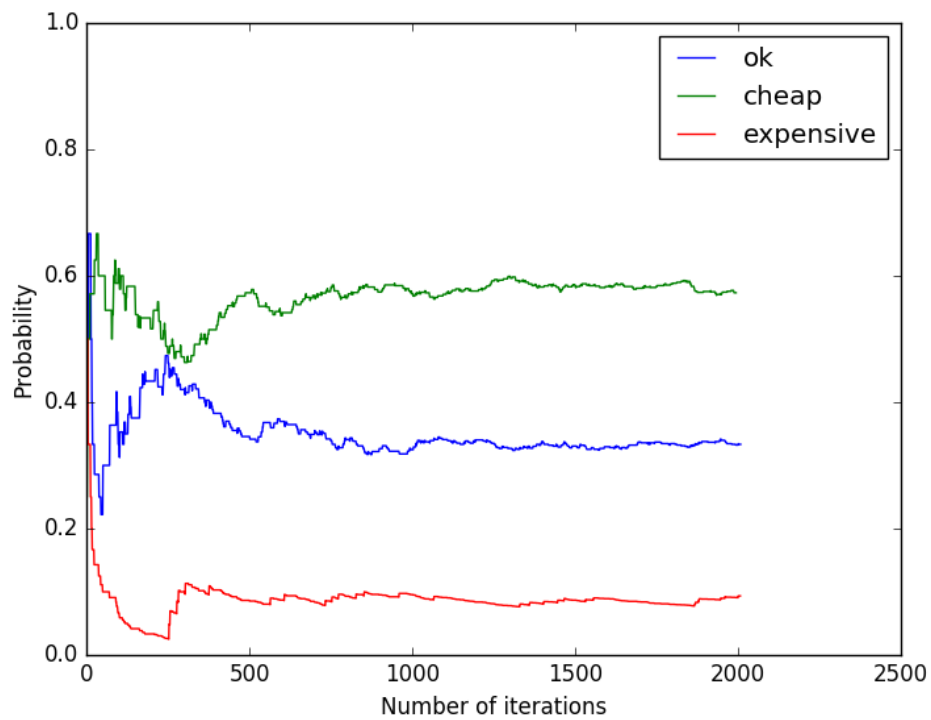
Number of iterations = 1000

Number of dropped samples = 0



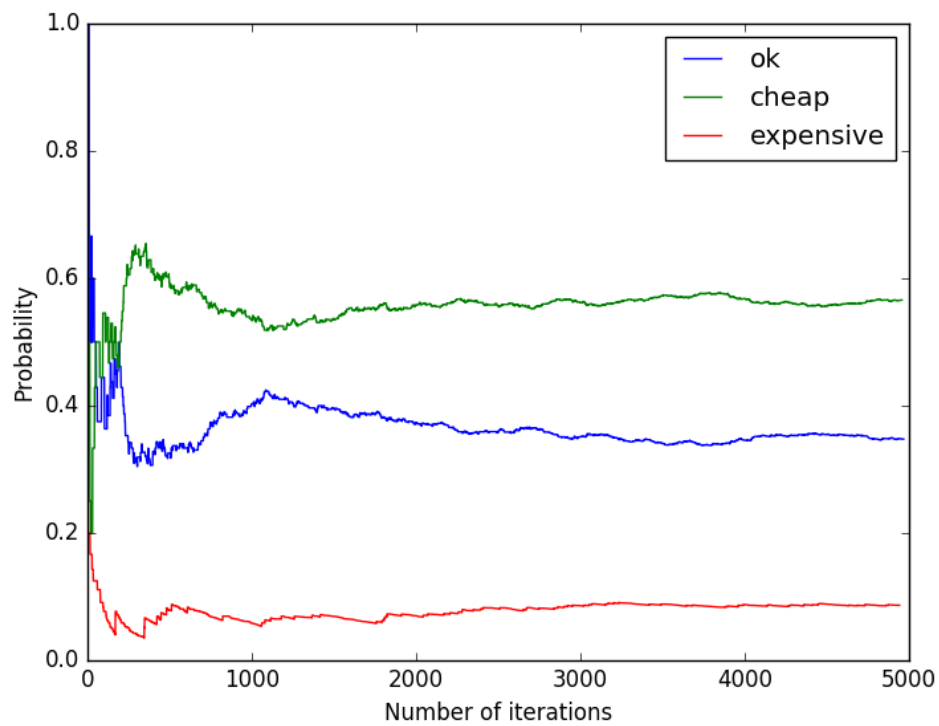
Number of iterations = 2000

Number of dropped samples = 0

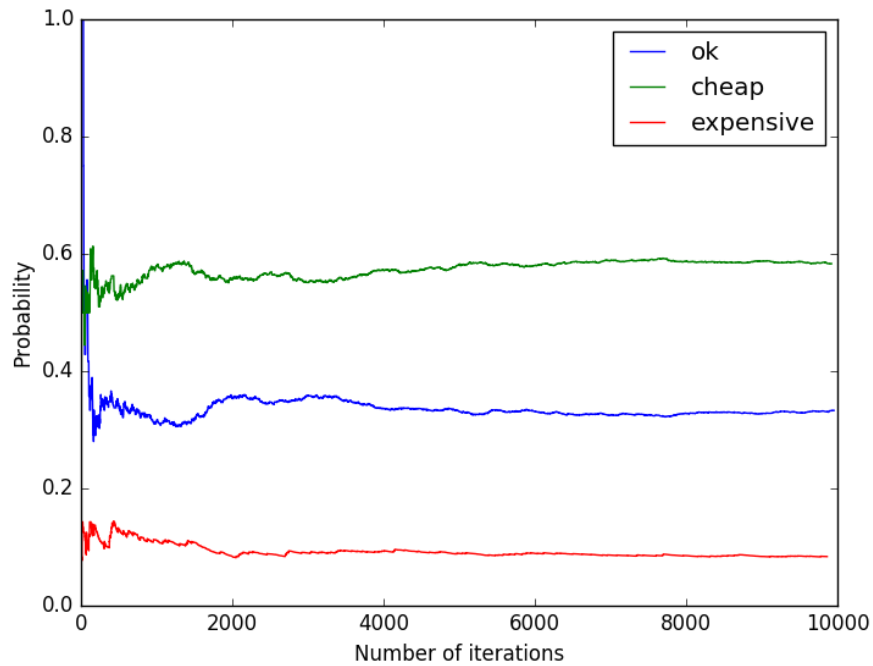


Number of iterations = 5000

Number of dropped samples = 0

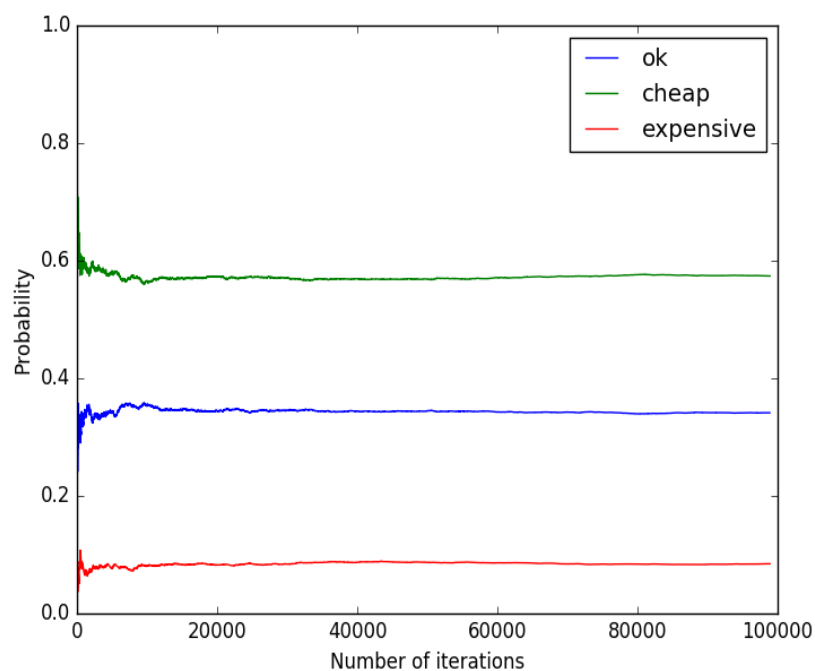


Number of iterations = 10,000
Number of dropped samples = 0



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
```

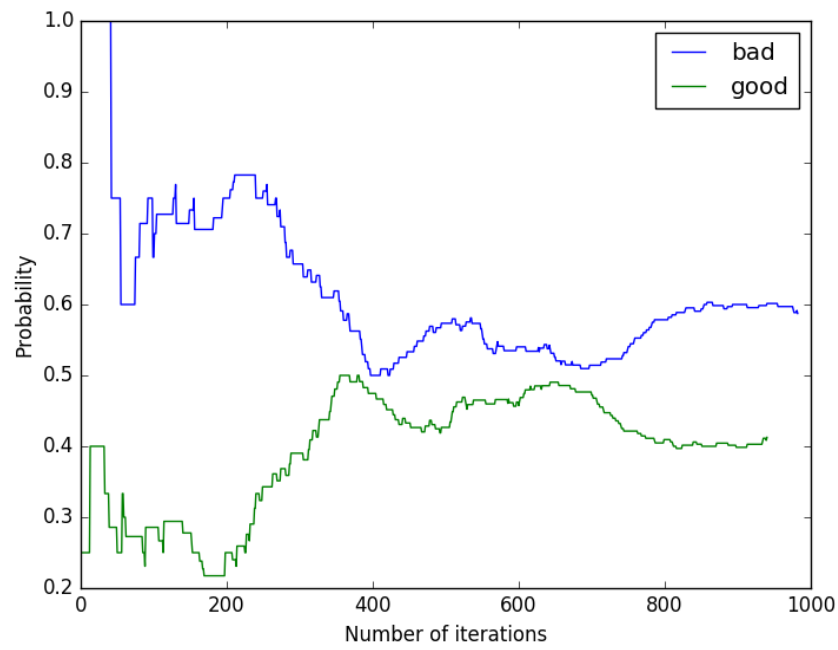


b) Query node - children

Evidence node - neighborhood = bad

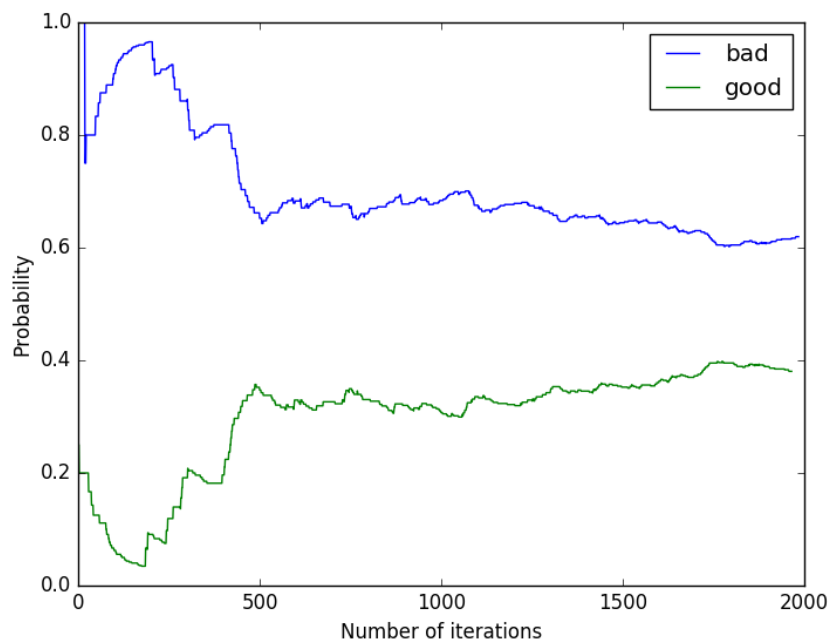
Number of iterations = 1000

Number of dropped samples = 0



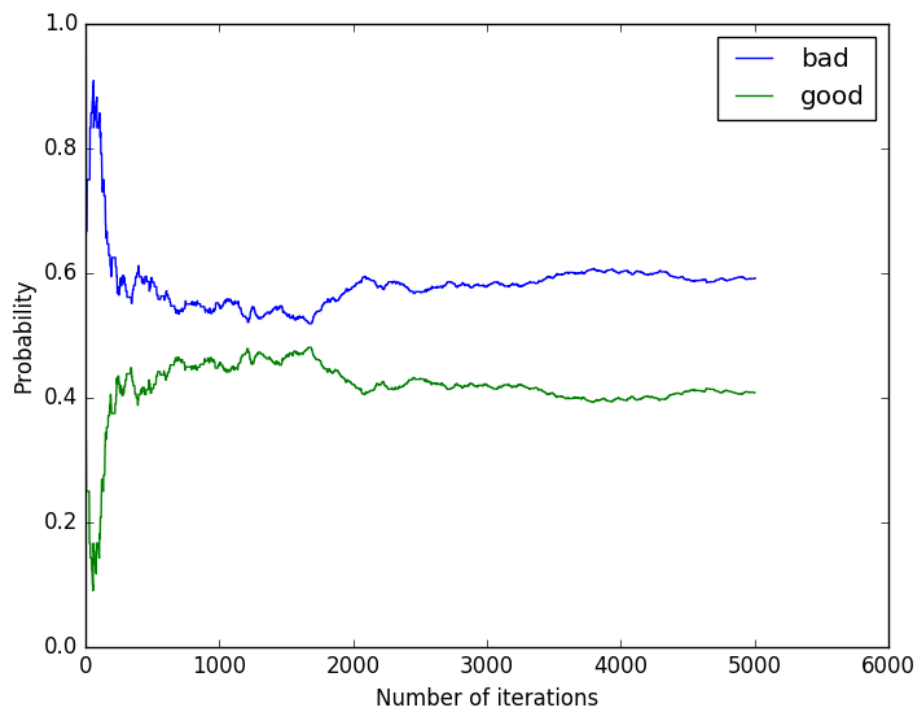
Number of iterations = 2000

Number of dropped samples = 0



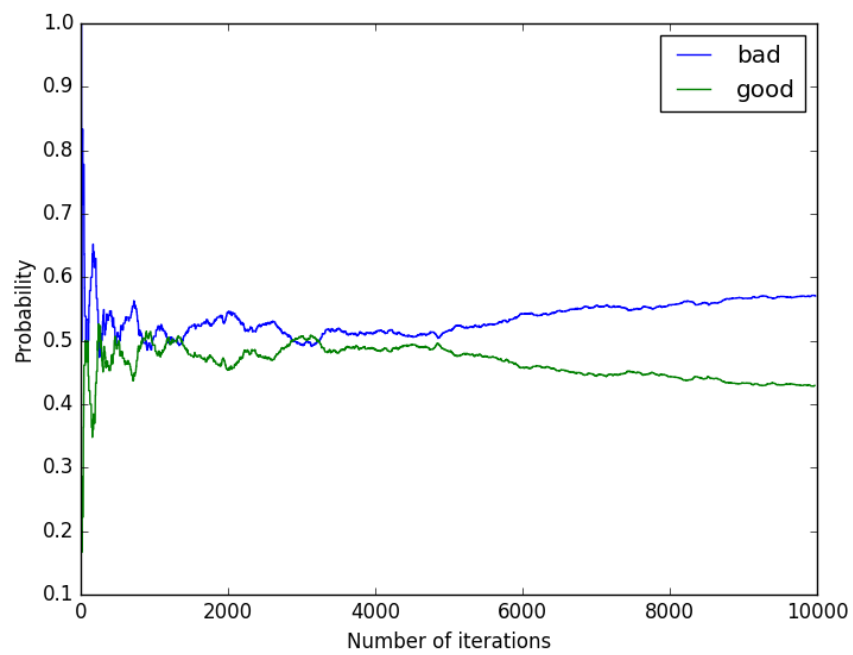
Number of iterations = 5000

Number of dropped samples = 0



Number of iterations = 10,000

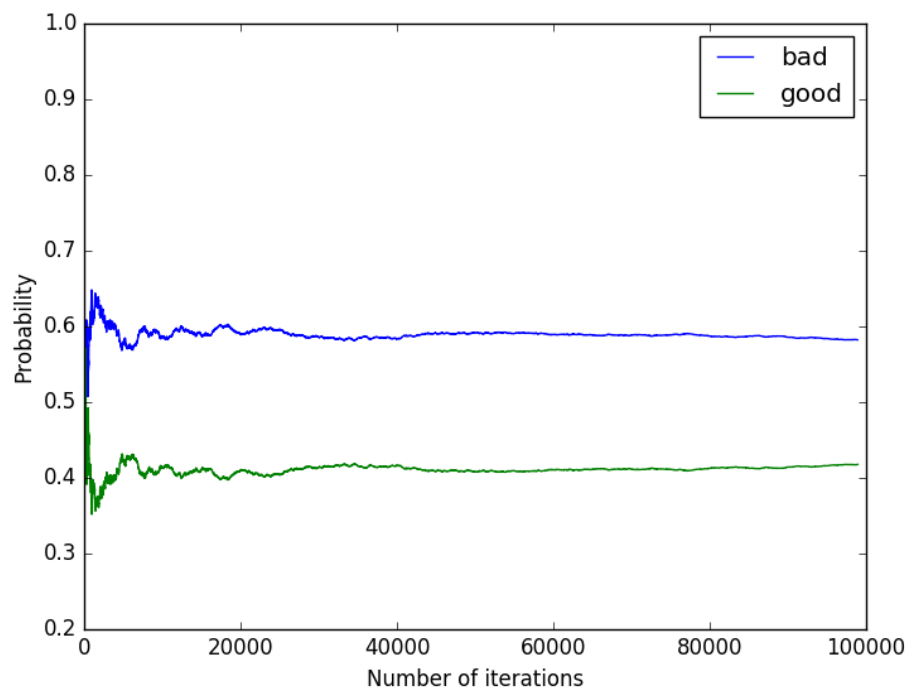
Number of dropped samples = 0



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
```



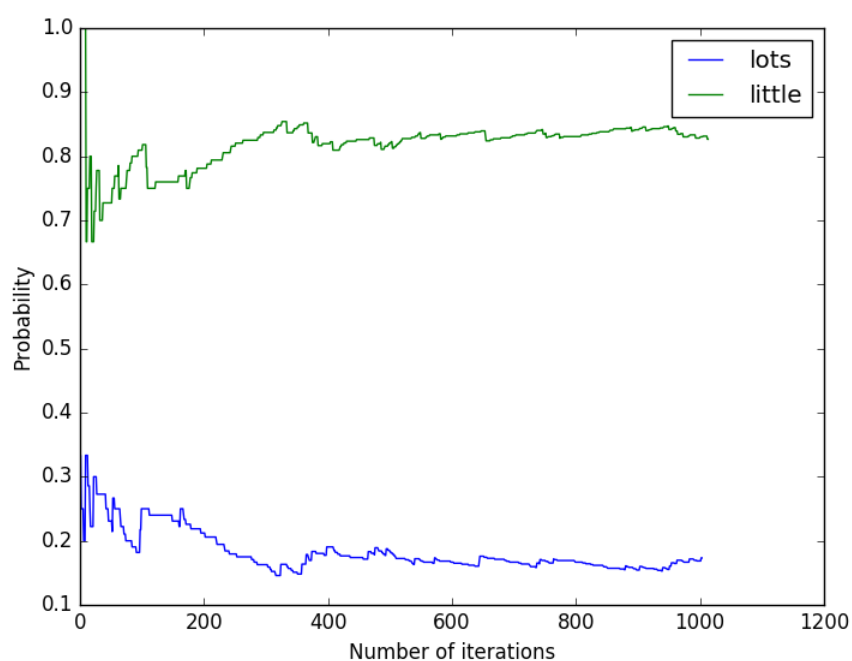
c) Query node - amenities

Evidence node - neighborhood = good

location = bad

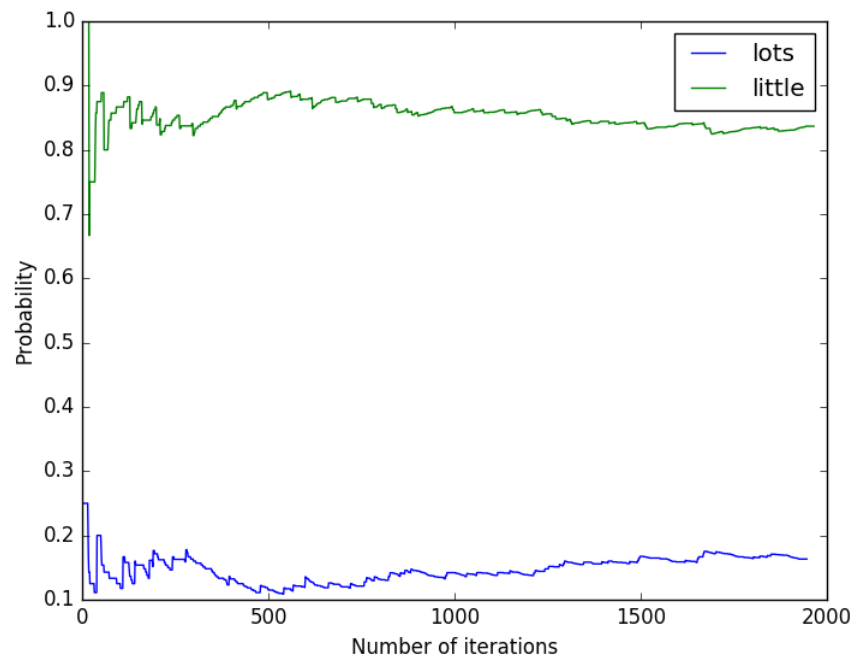
Number of iterations = 1000

Number of dropped samples = 0



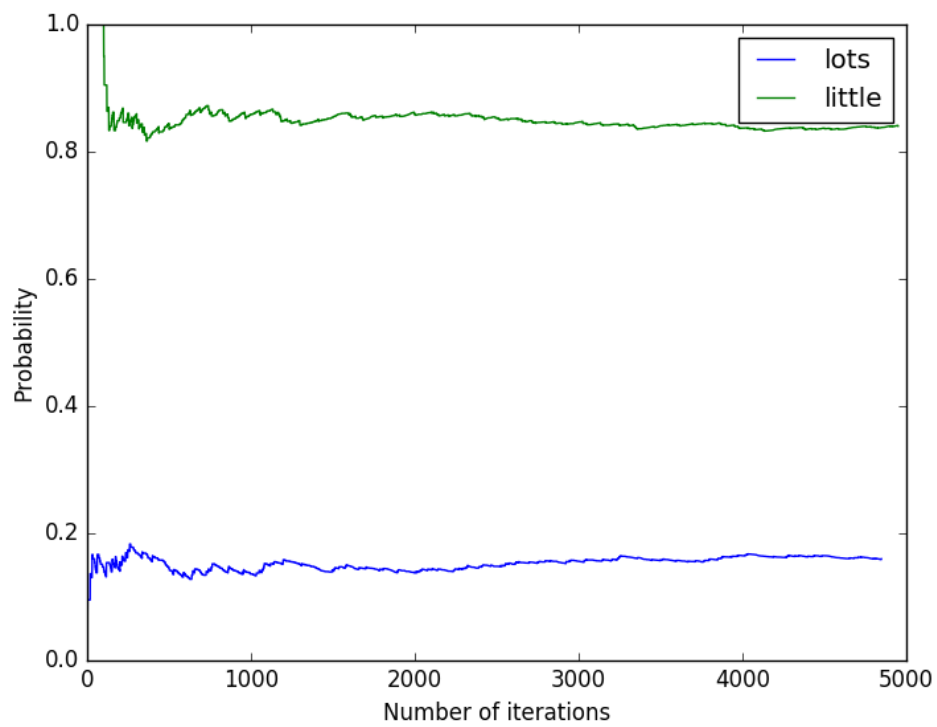
Number of iterations = 2000

Number of dropped samples = 0

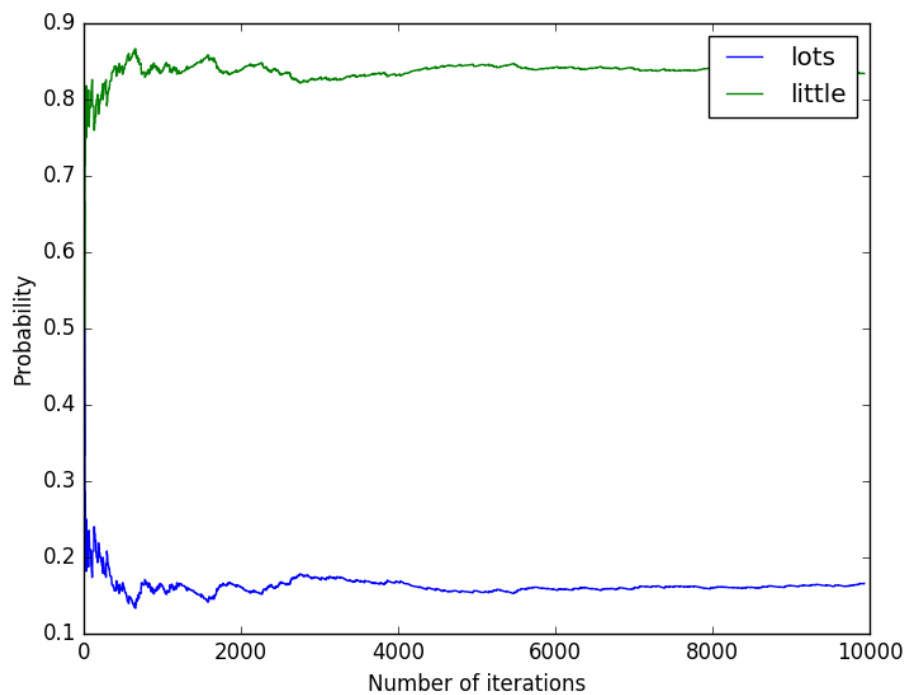


Number of iterations = 5000

Number of dropped samples = 0

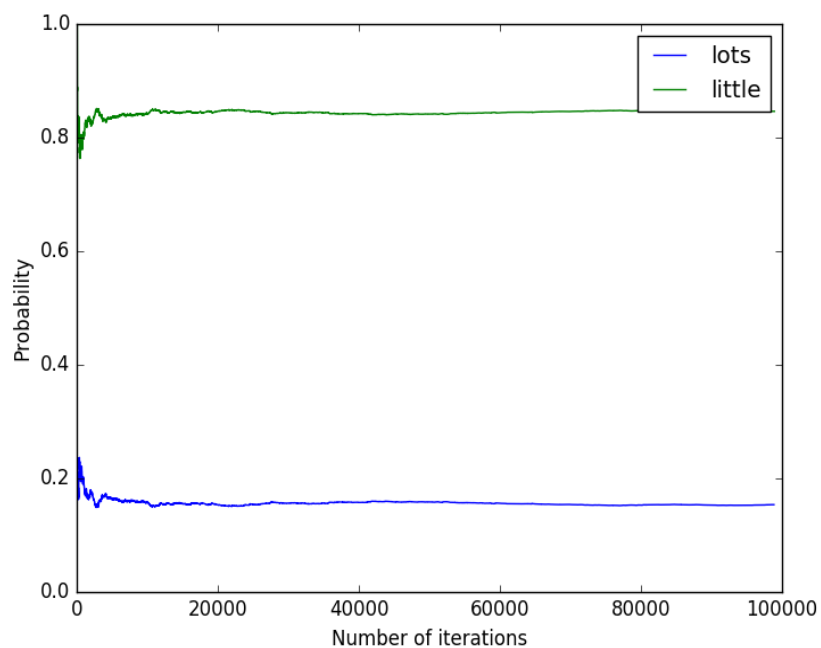


Number of iterations = 10,000
Number of dropped samples = 0



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
```



d) Query node - size

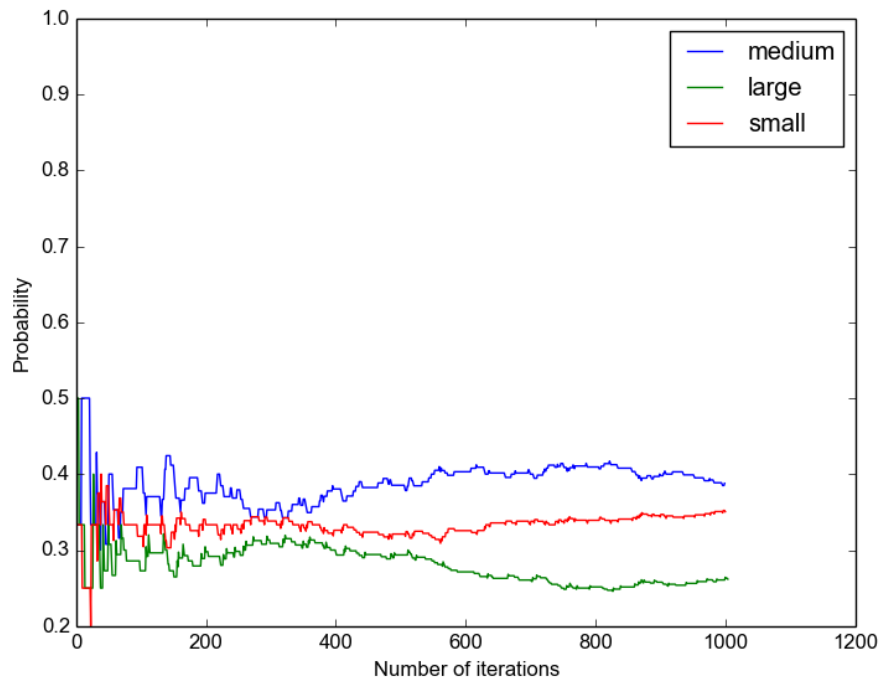
Evidence node - amenities = lots

neighborhood = good

price = cheap

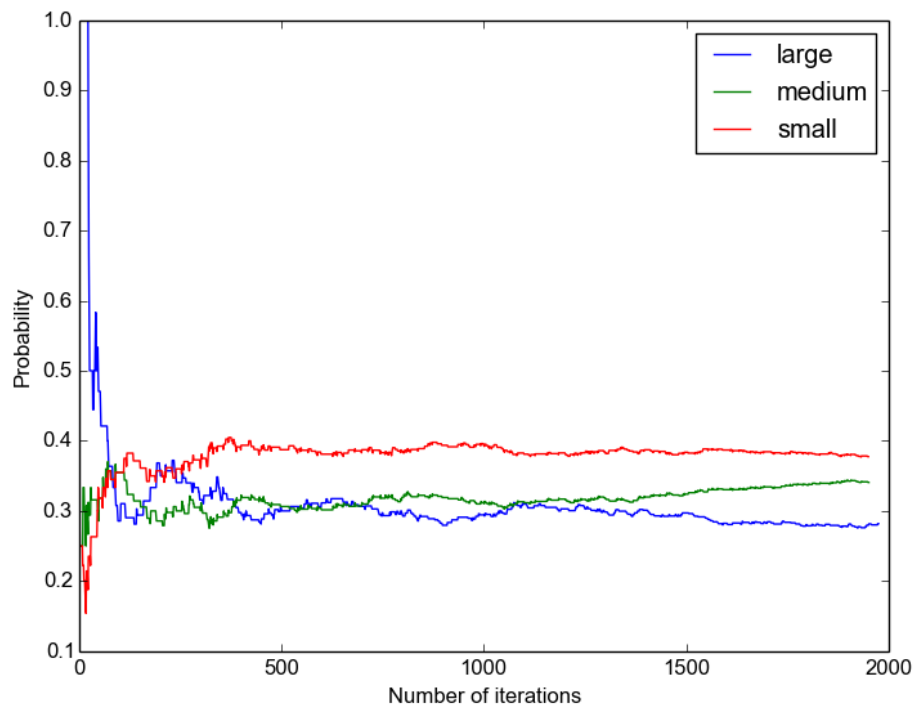
Number of iterations = 1000

Number of dropped samples =0

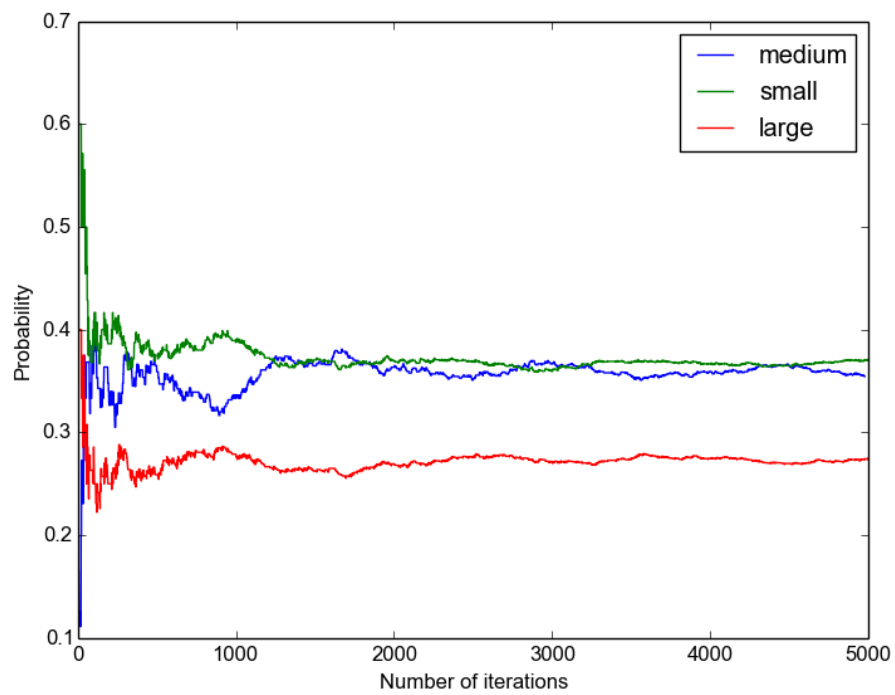


Number of iterations = 2000

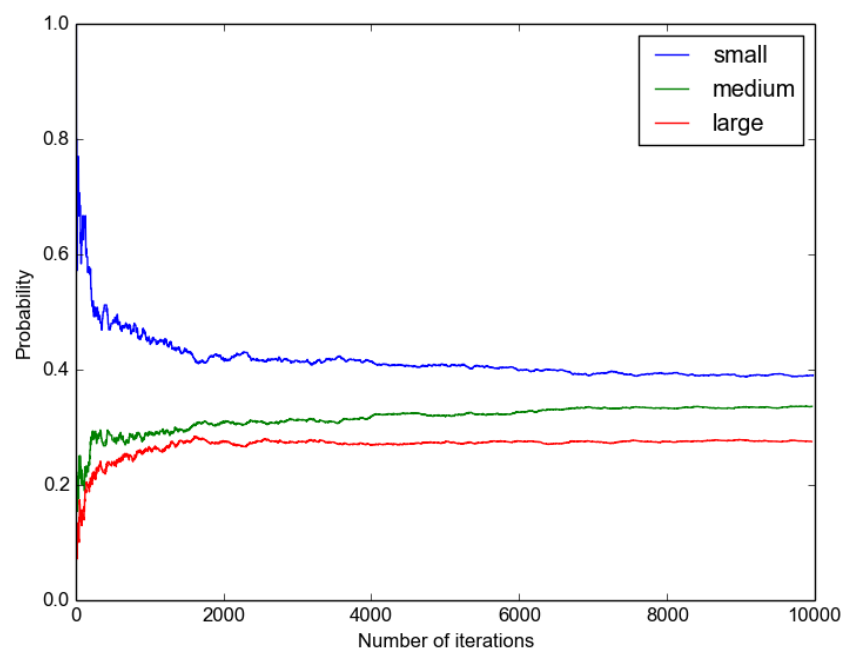
Number of dropped samples =0



Number of iterations = 5000
Number of dropped samples =0

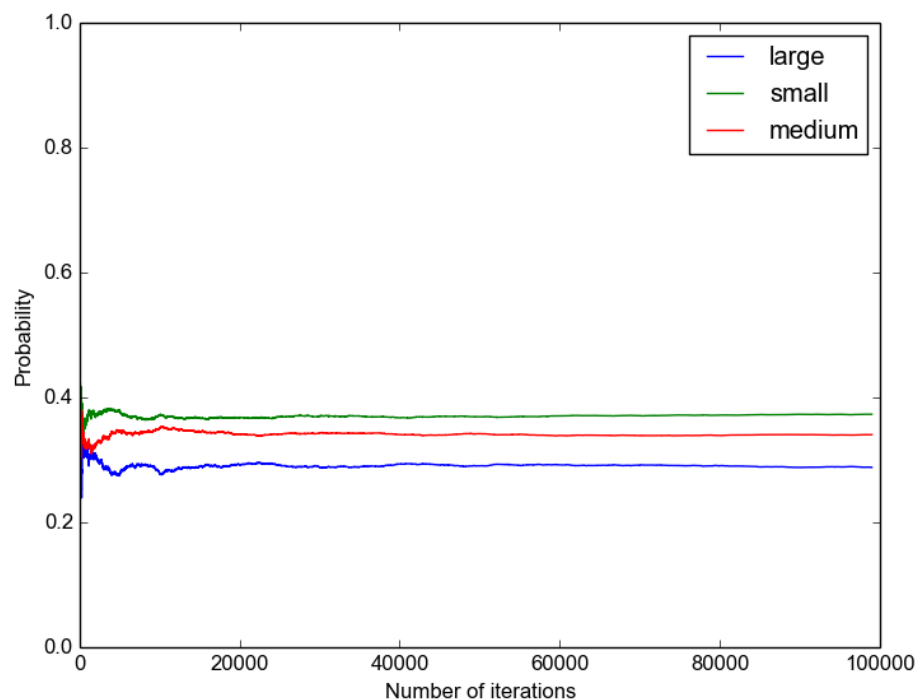


Number of iterations = 10,000
Number of dropped samples =0



Number of iterations = 100,000
Number of dropped samples = 0

```
nishant@Nishant-Inspiron:~/AI Assignment/Assignment-2/AI/Project 2$ python3 Gibbs.py size amenities  
=lots neighborhood=good price=cheap -u 100000 -d 1000  
P(size=large) = 0.28753977473609776  
P(size=small) = 0.3725440678822163  
P(size=medium) = 0.33991615738168596  
Time taken for 100000 iterations is 2.9441728591918945 s
```



The mean number of iterations it takes to converge for the cases mentioned above is around 5000. It is not the same for every node queried. It takes lesser number of iterations to converge for a few nodes and for a few it takes more than 5000 iterations (for queries with one evidence nodes).

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.

Say you have 8 nodes. For one query say we have the number of evidence nodes as 1, i.e., $P(A|B)$ and for another one we have number of evidence nodes as 2, i.e., $P(A|B \text{ and } C)$. In both cases we assign the evidence nodes and then we randomly assign the other nodes. In the first case we can see that there are 7 remaining nodes over which we sample and assign states to, based on probability computed over its Markov Blanket and in the second case there are only 6 (this is because the evidence nodes never change). Say we go through the nodes in a sequential manner, i.e., we go through the tree entirely once assigning nodes and then only we assign a node that has been already assigned. The query node probability will change each time the nodes in its Markov blanket are differently assigned. Say we sample 42 times. In the first case (with 1 evidence node) we can see that the query node has been assigned only 6 times ($42/7$), but in the second case the query node has been assigned 7

times (42/6). Say the probability for any query node converges when the query node has been visited around 2000 times. The number of iterations for this in the first case ($P(A|B)$) would be around 14000 (7×2000) samples and in the second case it would only be 12000 samples. Therefore, we feel the number of samples required for convergence depends on the number of evidence nodes.

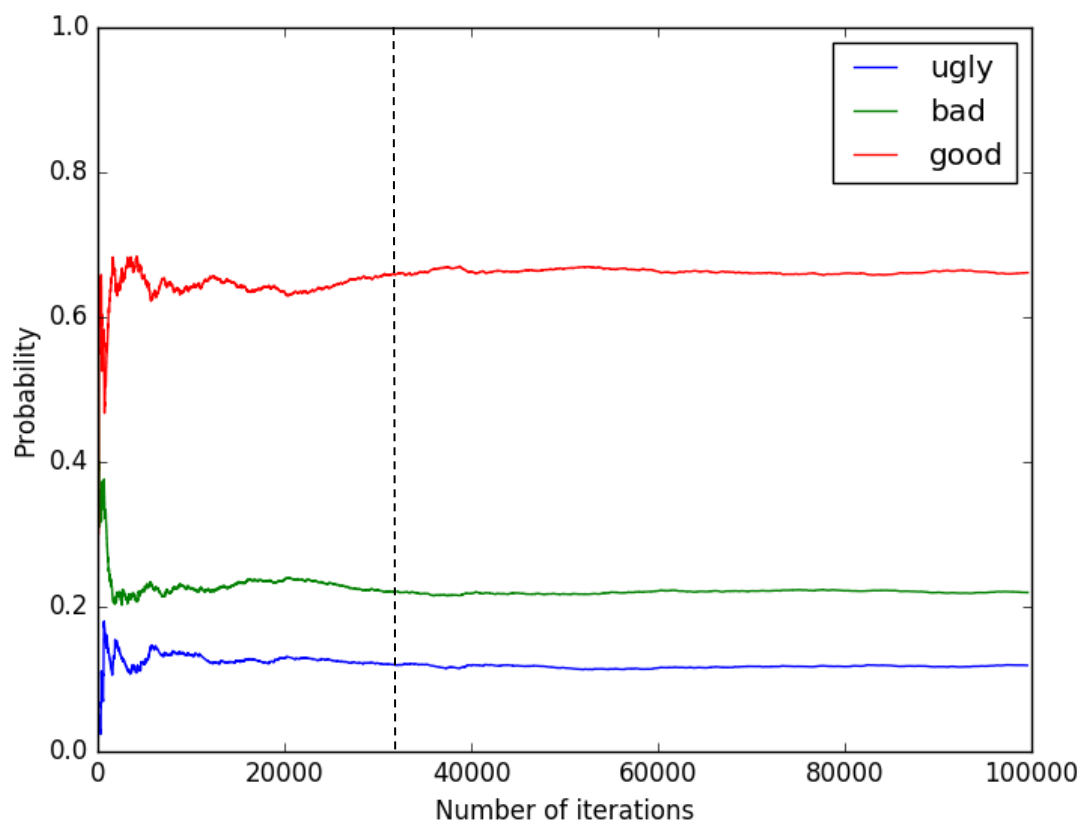
This is in congruence with the graphs we have plotted in the previous question. For a query with just one evidence node it takes a lot of iterations to converge. But for a query with 2 evidence nodes it takes far lesser number of iterations to converge.

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

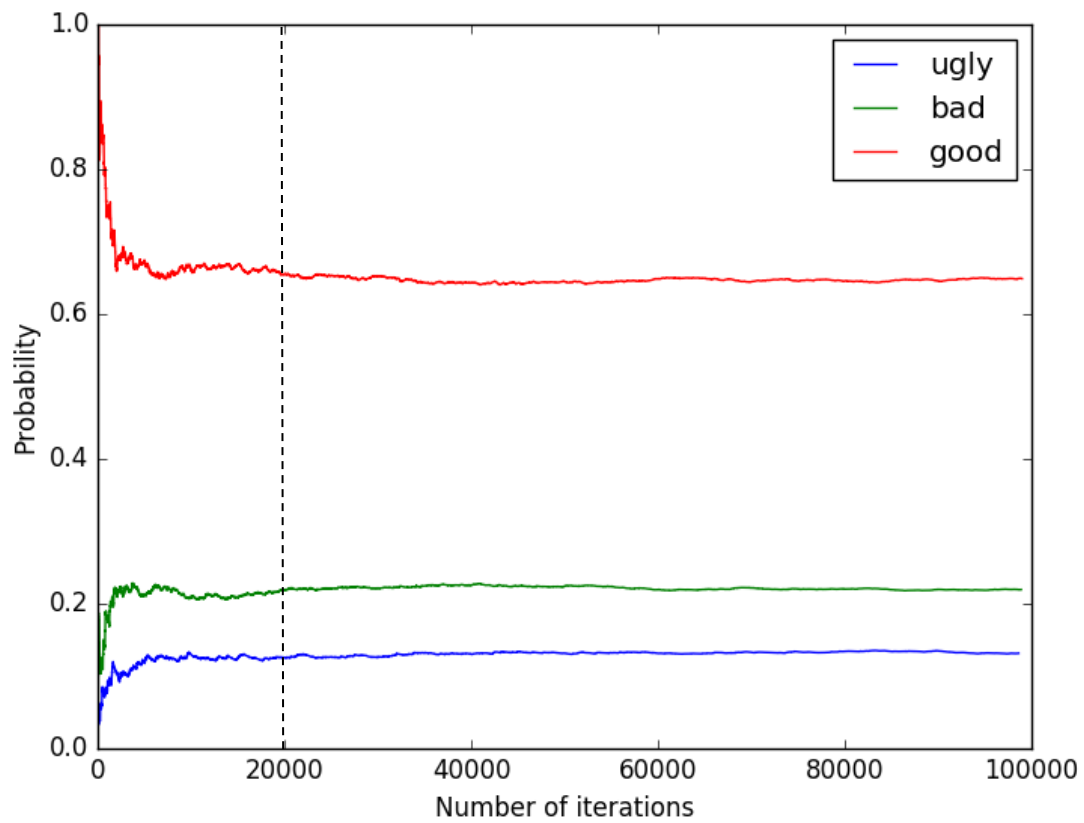
ANSWER:

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