Collaborative Filtering : Midsem Exam

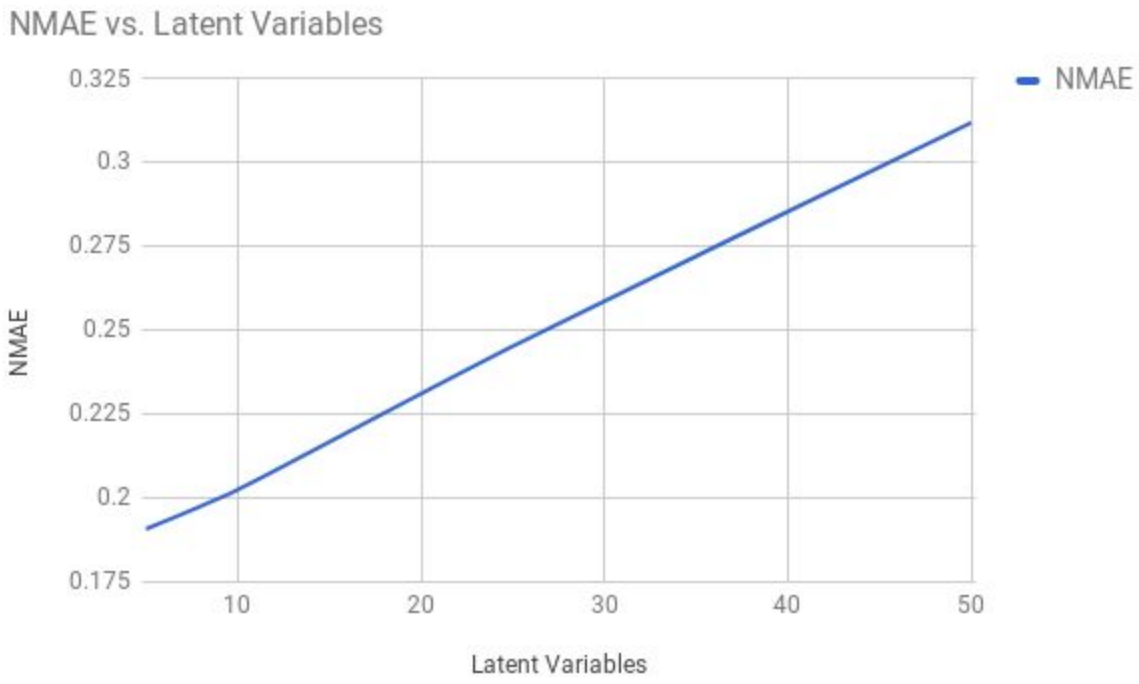
Anshuman Suri : 2014021

Note: Implementation is attached in submission. Run *bash download.sh* to acquire dataset. Run the code by running *python latent.py ml-100k/*.

Grid search was run over the number of hidden latent variables. For a given value of lambda, it was observed that the *NMAE* increases as the number of latent variables is increased. After running the algorithm over [5,10,25,50] latent variables, 5 was used (as it gave the lowest *NMAE*).

```
anshuman@ultron:~/Desktop/Studies/CF/Assignments/Midsem$ python latent.py ml-100
k/
('For', 5, 'average NMAE over all folds:', 0.19082548782177175)
('For', 10, 'average NMAE over all folds:', 0.20259573054629026)
('For', 25, 'average NMAE over all folds:', 0.24525453949698481)
('For', 50, 'average NMAE over all folds:', 0.31199621614241158)
```

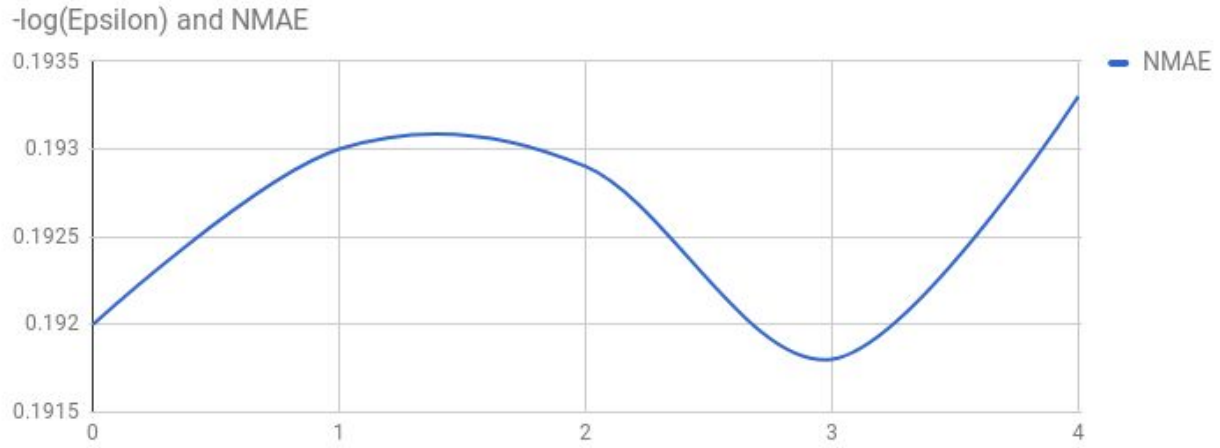
Here are the results after running for varying values of number of hidden variables,for a value of epsilon as 1e-1.



After selecting this, grid-search was run again over values of lambda *[1,1e-1,1e-2,1e-3,1e-4]*. As it can be observed, the minimum *NMAE* was achieved for lambda value as *1e-3*

```
anshumans@ubuntu:~/Midsem/Midsem$ python latent.py ml-100k/
('For latent', 5, 'and lambda', 1, 'average NMAE over all folds:', 0.19206845318463789)
('For latent', 5, 'and lambda', 0.1, 'average NMAE over all folds:', 0.19307212553091788)
('For latent', 5, 'and lambda', 0.01, 'average NMAE over all folds:', 0.19288836993392583)
('For latent', 5, 'and lambda', 0.001, 'average NMAE over all folds:', 0.19183119345363711)
('For latent', 5, 'and lambda', 0.0001, 'average NMAE over all folds:', 0.1932544970110347)
anshumans@ubuntu:~/midsem/Midsem$
```

Here are the results, after running for varying values of epsilon for 5 (chosen by grid search) latent variables.



Thus, the final configuration works best when it has 5 hidden latent variables, and lambda as .

Thus, the final optimal *NMAE* comes out to be **~0.19**, when running for 5 hidden latent variables and having an epsilon value of 0.001. All experiments were run with 5 fold cross validation, training with 100 iterations (without early stopping).