Report on Assignment 3 : BKT Model

Description of Code Structure :

PyCharm, an IDE for python was used for implementing the code. There are 2 sources for the code :

1. First .py file is for Single KC's which shows 7 outputs for 7 individual KCs
2. Second .py file for Multiple KC which gives one output that includes Learning probability, correctness probability for 7 individual KSs and Multi KC learning probability and correctness value for Multi KC.

USE of this models :

When given large data sets corresponding to student learning behavior of different concepts, this type of model can be used.
We can Refactor the code according to the Number of concepts included in the training set and the model would give us appropriate results.

Implementation :

Per KC model :-

- The code will first read the data from file which has 36678 records. Each column from the file is put into an array so that a list is made from 10 different arrays by zipping them.
- Seven empty arrays were created for seven KC's.
- Only if the student has used that particular KC, that record will be added to that particular KC's array.
- Now these arrays are sent to 2 functions, one to return the learning probability value that student has learnt the concept(KC) and the other

function return the correctness probability value i.e if the student has succeeded in that step.

- Once these functions return arrays, they are again zipped to that particular KC.
- So we now have the student record followed by 2 other columns : Learning probability value and Correctness probability value.

Multi KC model :-

- Here all the data from the file is split column wise and appended to array.
- These arrays are zipped to produce a list.
- This list is sent to two functions. First function to calculate likelihood value and second function to calculate correctness value.
- Now we again zip the data with 14 of the returned arrays from the previous function. (7 arrays for likelihood and 7 for correctness).
- Now this list is sent to a new function which calculates Multi KC correctness.
- Multi KC correctness is based on correctness of individual KC's when a step uses multiple KCs.
- The below equation id for individual KC correctness value.

$$P(C) = [\ P(L) * (1-P(S))\ ] + [\ (1-P(L)) * P(G)\ ]$$

P(C) = Probability that answer is correct
P(L) = Probability that student learned the concept
P(S) = Probability that student has slipped the answer
P(G) = Probability that student guessed the answer

- Generalizing this, the following formula was implemented if the user worked on multiple KCs.

$$P(L') = \sum R_i * P(L_i) \quad \text{if } P(L_i) \neq N/A$$
$$P(C) = [\ P(L') * (1-P(S))\ ] + [\ (1-P(L')) * P(G)\ ]$$

Where R_i values correspond to Relevance factor.

A relevance factor is assigned to a KC depending upon how many times that KC was used by the students. (Frequency of that particular KC)

Sources for Parameter Setting :-

Corbett and Anderson papers were used as reference to develop this model and set the default parameters ($P(L_{in})$, P(S), P(T), P(G))

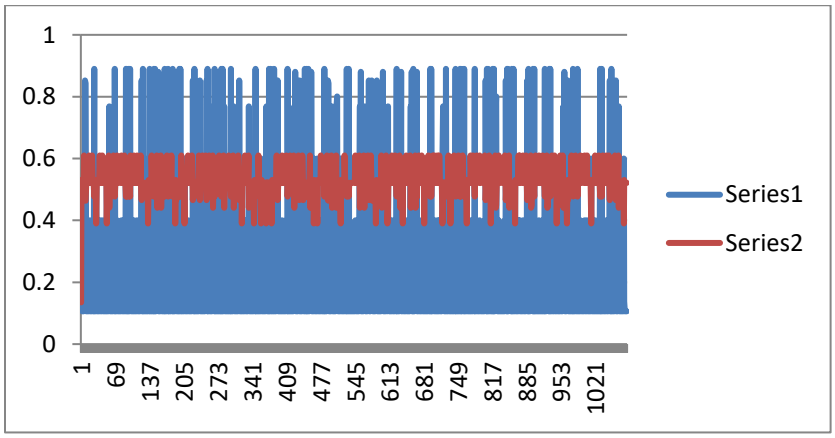
Performance :

Multi KC performs better than Single KC as per the following analysis. (This may change for individual per KCs.)

To compare the performance of Single KC with multiple KC, the error values of correctness probability are compared.

Error value is $\sqrt{(actual-observed)^2}$

We square root the squared value to avoid negative values in the analysis.
The KC1 error values and Multi KC error values are compared and the graph below is it's result.
Observing the graph, it can be traced that Multi KC performed well than Single KC.



Series-1 : Single KC (KC1)
Series-2 : Multi KC