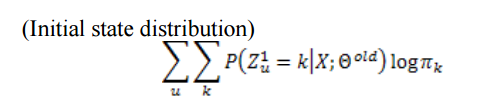
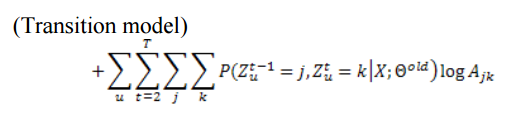
1.The starting probability distributions over the latent classes for each user (π).

* + Initialize an ArrayList with the starting length of k (where k is the number of states).
  + Fill in counts of users who start in each state.
  + How do I assign initial states

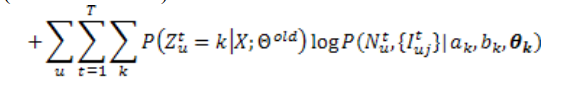


2. The transition probability table between classes in adjacent time periods (A).

* Create forwards and backwards state moving methods and call based on probability of state change
* How do I determine when to change states?
* Alpha is the forward variables
* Beta is the backwards variables



3. The emission or observation model that generates the data from the latent class memberships in each time period



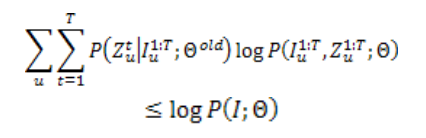
//Optimization Step

//Maximization/M-step: The parameters are calculated

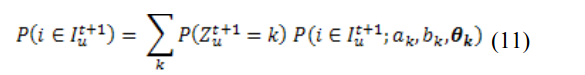
//such that, for a given distribution over the hidden variables,

//the expected log likelihood of the parameters is

//maximized.



Prediction Step



1. Use data collected over time period 1 … ttrn for training.

2. Initialize π, A, (a, b), θ to small random values.

3. E-step: Compute P(Zt u|I 1 u :T) and P(Zt u –1, Zt u|I 1 u :T) using equations (2) and (3).

4. M-step: Estimate π, A, θ using equations (7), (8), and (10). Estimate (a, b) using Section 2.1 of Minka (2002).

5. If expected log likelihood has not converged, go to step 2.

6. For each user u compute R(i, u) of each item i for time period ttrn + 1 using equation (12). a. Recommendation top N items with highest R(i, u).

