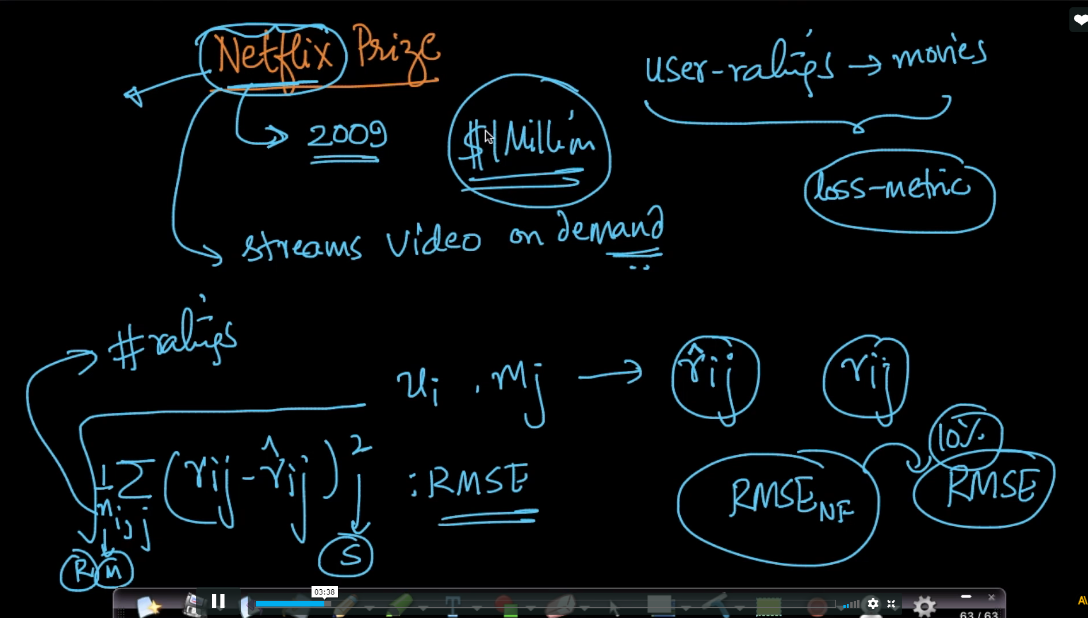
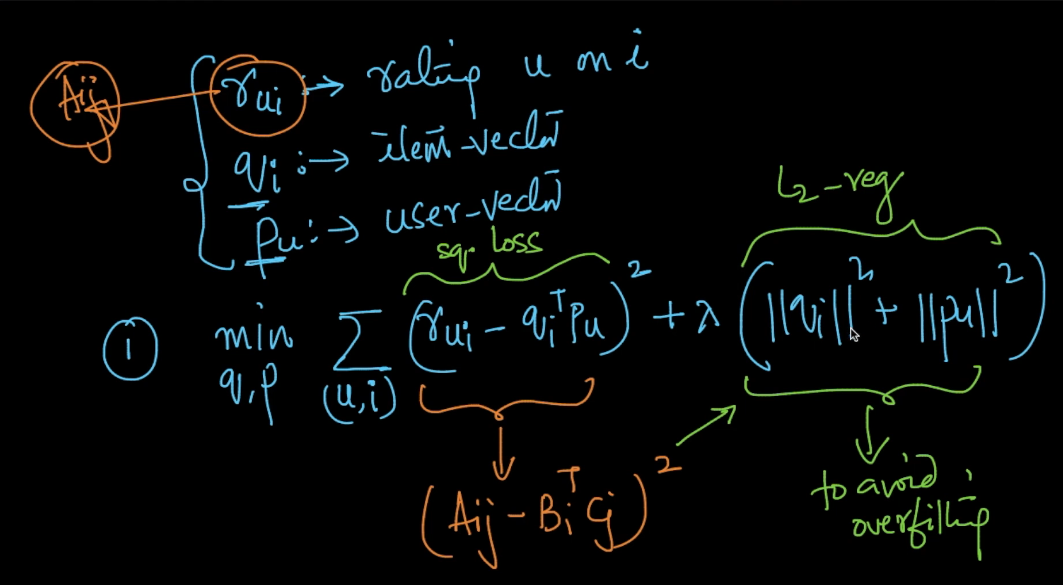
**Matrix Factorization for recommender systems: Netflix Prize Solution**



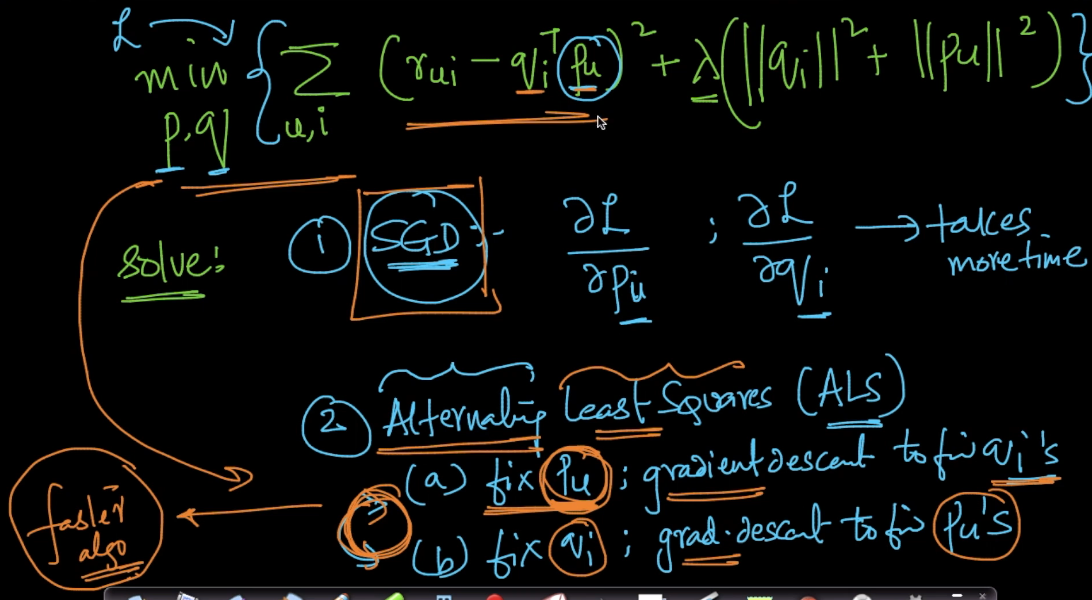
Below image show notation used in research paper and optimization equation here it also uses regularization term to avoid overfitting



To solve this optimization problem we use SGD but SGD tooks times therefore we use alternating least squares(ALS)

In ALS first we fix Pu (keep Pu constant) and we use gradient descent to find qi’s

Then we fix qi (keep qi constant) and we use gradient descent to find Pu’s



Now in this formulae we use 3 new terms

Mu : this is average of rating means average rating across all users & item ecosystem(here netflix)

Bi : bias term for item means how much it is different from other items but don’t consider users

Bu : bias term for users means how much it is different from other users but don’t consider item

Above all 3 are scalar values

PuTqi : interaction effect means how much actually user feels for this movie

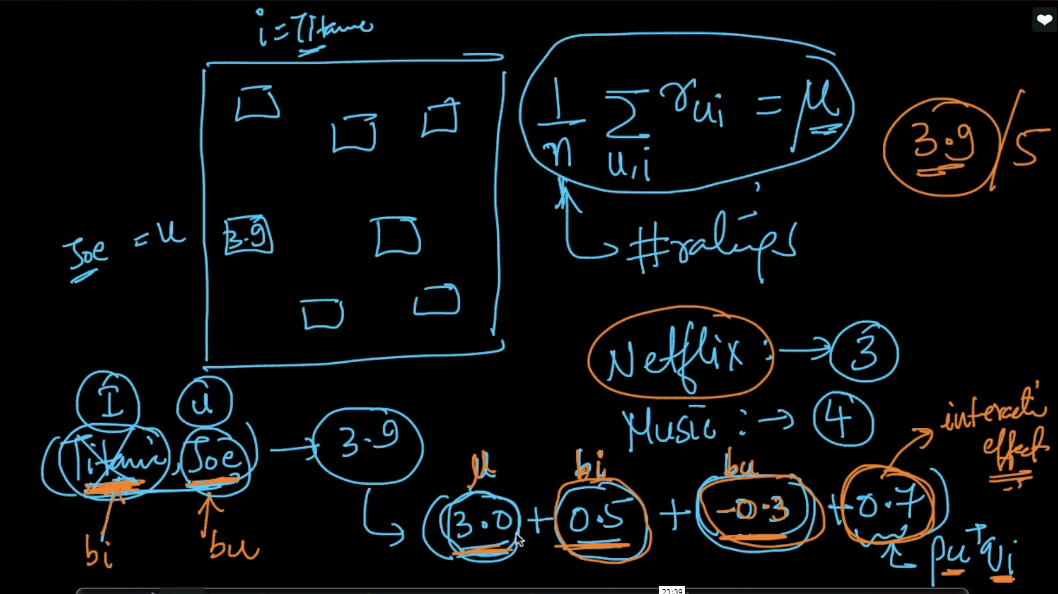
So according to this let we take example of movie titanic and user joe gives it rating 3.9

So in this mu\_ = 3.0 is average rating across all users and items ecosystem

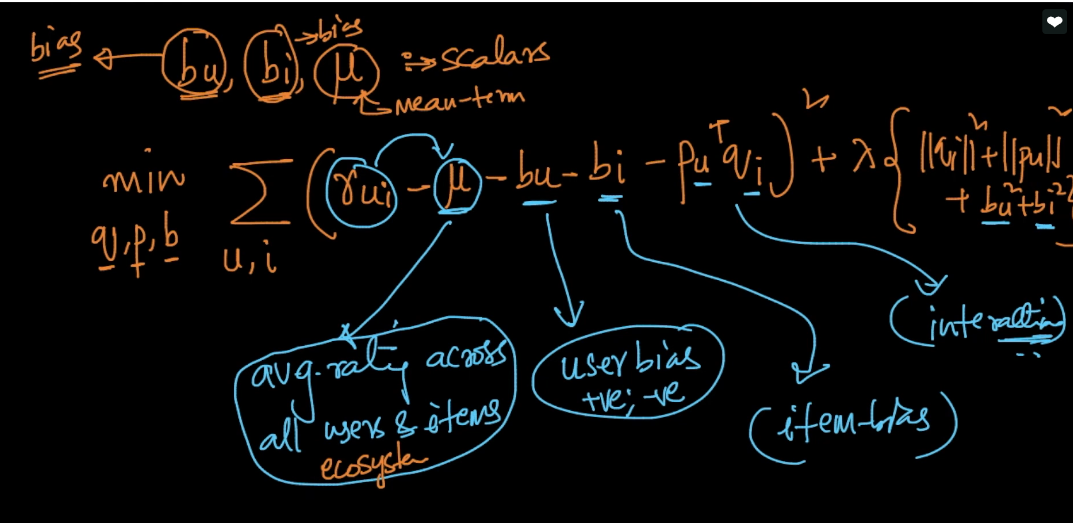
Bi = 0.5 because titanic is very popular movie therefore it is more bias.

Bu = -0.3 because suppose user joe usually gives low rating therefore it is less bias or negative

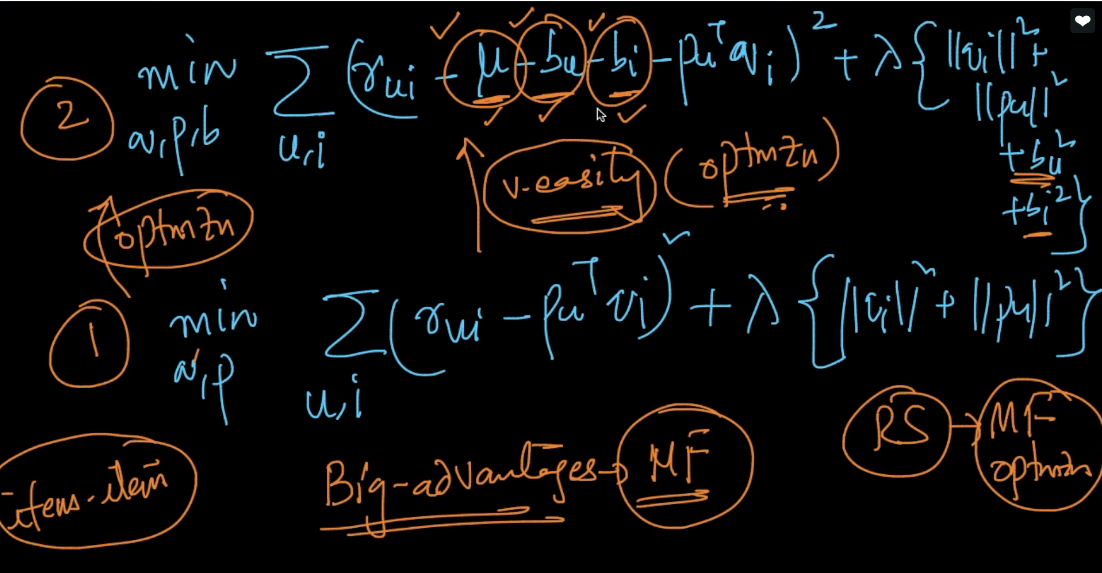
PuTqi = 0.7 this is actual rating which joe feels about titanic movie



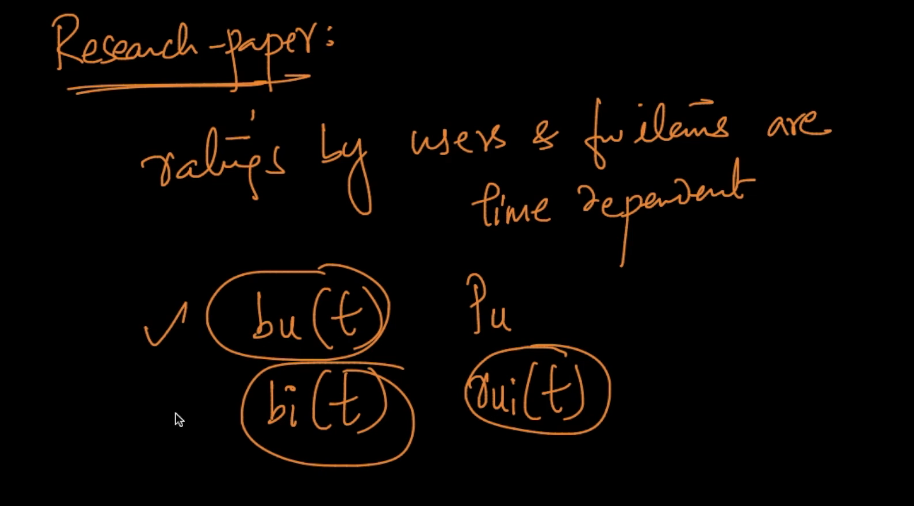
So after adding these new terms in optimization formulae new transformed formulae is shown below as we add these new terms therefore we also add bias terms in regularization to avoid overfit.



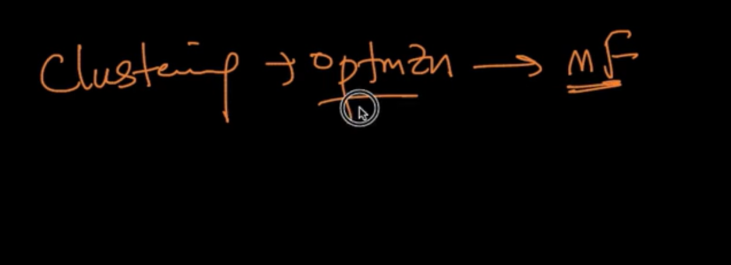
So here we see the big advantage of Matrix factorization is that we can add new terms easily using domain knowledge



In research paper these things are time dependent



Therefore we see how we convert clustering optimization into MF



Comments :

