Course Name - Recommender Systems
Professor Name - Prof. Mamata Jenamani
Department Name - Industrial and Systems Engineering
Institute Name - Indian Institute of Technology Kharagpur
Week - 08
Lecture - 36

Lecture 36: Hybrid recommender systems

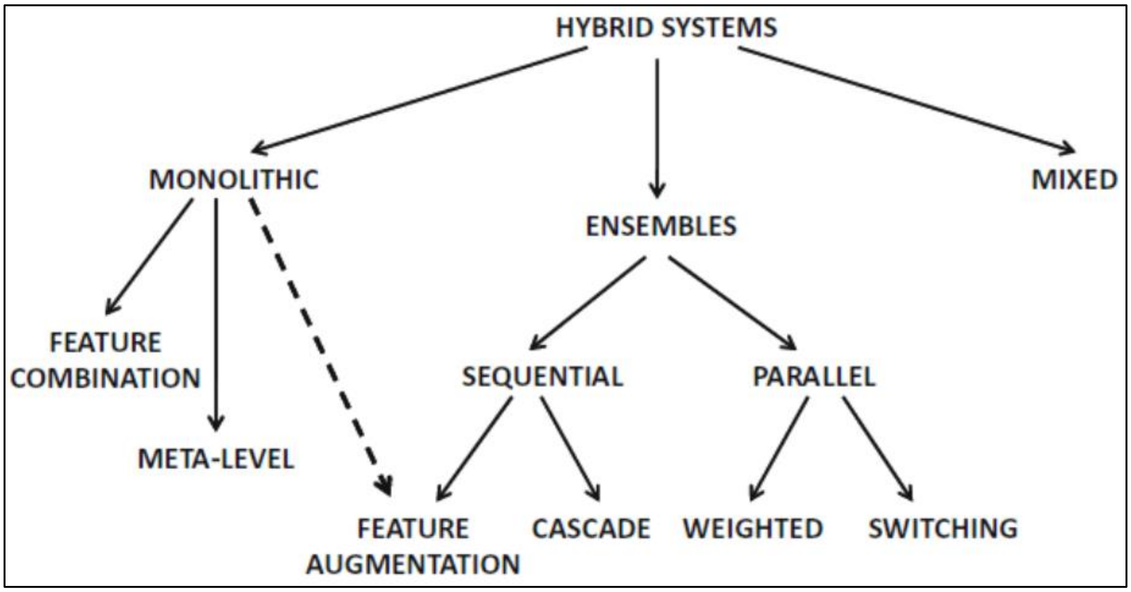
Welcome to the last module of this recommended system course and from today onward for next few lectures we will be talking about various other kind of recommender systems. So if you remember we have talked about only collaborative filtering and content based systems. But there are hybrid systems as well which not only combines the outcomes of collaborative filtering and content based filtering, the other systems which we are going to study they also combine those outcomes. So today we limit ourselves to how to hybridize from a number of different recommender systems. To start with these hybrid recommender systems are used either to leverage the power of multiple data sources or to improve the performance of the existing recommender system within a particular data modality. So here as the name indicates we do hybridization by combining multiple recommendation algorithms or use the data from multiple sources to enhance those algorithms.

So there are three important design strategies here, Ensemble design, Monolithic design and Mixed system design. As the name indicates in case of ensemble design we get the results from the existing algorithms and combine them to generate a generate the recommendations generate single recommendation which is more robust. In case of monolithic design, we integrate many recommendation algorithms to create one algorithm which may be using various different data types. In case of mixed system, it is almost like ensembles and here you use multiple recommendation algorithms as a black box but the items recommended by various systems are presented together side by side you do not combine the recommendations.

So this is the typical taxonomy of hybrid system. As I told you this is monolithic ensemble or mixed. So this we already discussed this we already discussed monolithic ensemble and mixed. Now there are few other new items that ensembles are of sequential or parallel and there are many other taxonomies which are which relates to different category of recommendations. So these taxonomies we are going to look at.

First one is weighted. Here the scores of several recommender systems are combined into a single unified score by computing weighted aggregate of the scores from individual ensemble components. This is weighted. So this is weighted. Then you have switching.

The algorithm switches between various recommender systems depending on the current need. Then you have cascade. In this case one recommender system refines the recommendation given by another process called boosting. Bagging boosting etc. the people who of course we did not discuss about them much but in ensemble kind of algorithms in machine learning this concept of bagging boosting all those stuffs come.



So this is something similar to boosting. Where is cascade? Cascading. Then feature augmentation. Here the output of one recommender system is used to create the input feature for the next one. Feature augmentation.

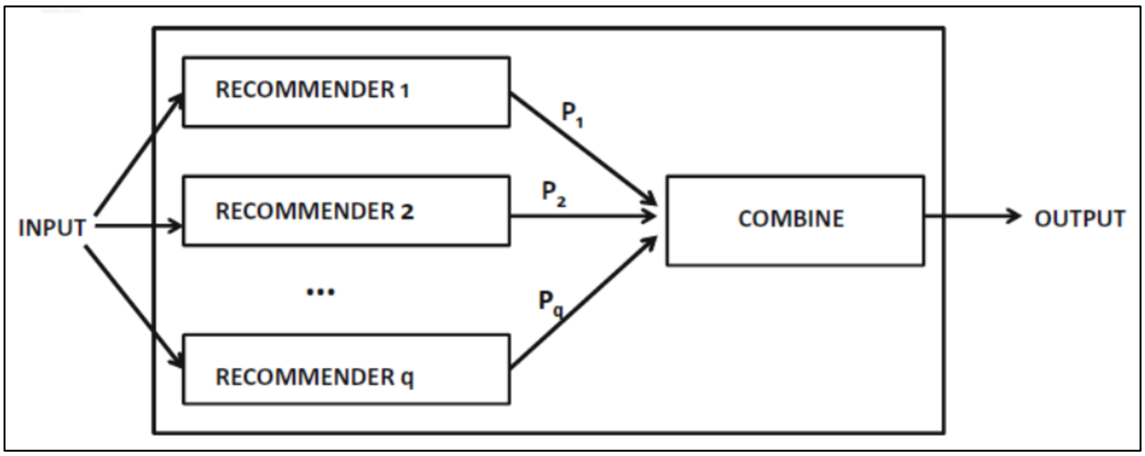
Then feature combination. The features from different data sources are combined and used in the context of a single recommendation system. Feature combination. Then you have meta level. The models used by one recommender system is used as the input to the another recommender system.

Then you have this mixed. Mixed of course we have already discussed. The recommendations from several engines are presented to the user at the same time. So we had three approaches. First one was ensemble based.

This ensemble based as we saw in this diagram in the previous diagram it can be either it can have either parallel design paradigm or it can have sequential design paradigm. In case of parallel design paradigm various recommendation functions independently of one another are used and predictions from the individuals are combined at the end. So this from individuals the recommendations are combined. Now how these combinations take place has to be can be different. There can be weighted method in which we join them together using certain importance weights or it can be switching method.

So in case of switching depending on the situation results from a particular recommender algorithm will be shown. But in case of sequential design the output of one recommender is used as the input to the other. So first one, this one. So in a cascaded manner it is it happen. So the cascade and meta level systems can be viewed as examples of sequential methods.

Why this ensemble method based work? The theory behind this is this bias variance theory. Error typically though we did not discuss I just like to mention typically the error total error is error contributed by bias variance and the noise. What is bias? Bias is the component which comes from the modeling assumptions of the classifier. It has a linear decision boundary or it has to follow certain rule and so on. Variance comes from the random variation in choices of the training data which results in inconsistent predictions.

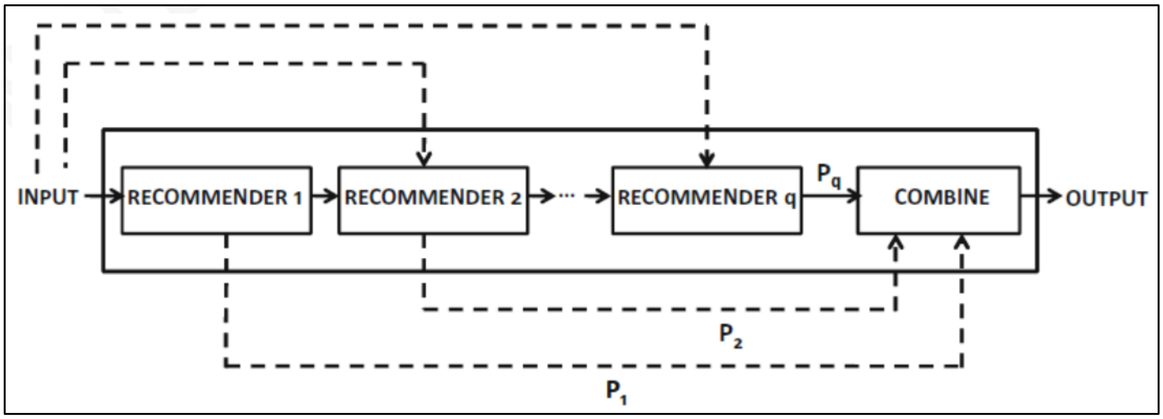


Then last one is your noise. The noise refers to the intrinsic error in the target class labeling which of course cannot be removed. So we which we can now have to play around in case of ensemble methods we can play around with bias and variance. Now since the experimenter do not have any control over the data quality that is noise by reducing the bias the process of reducing the bias one of the process is called boosting. Similarly, the process of reducing variance is called bagging.

So reducing this bias or variance one can reduce the overall error of the classifier. Now in case of weighted hybrids when exactly are you using weighted hybrids? Here also we saw combine in parallel. So we can use the weighted hybrids to combine them. So while combining them you will have recommendations from multiple engines. So this recommendation is in terms of the rating matrix.

So you will have this rating matrix generated from let us say Q algorithms. Now we are having to assign certain weights alpha 1 to alpha Q for each of this rating. So your R u j hat what is R u j? It is the rating value predicted for item j on behalf of item u. So item u would have made R cap on item j R u j cap on item j. So this matrix we got fully specified by one Q recommendation algorithms.

So you multiply this with the corresponding weight. So individual this is the individual this thing. So in the simplest case this is possible to choose everything is 1 by Q. However, depending on your choice you can give greater importance to some of the algorithms. Now when you combine the models using weighted hybrid there can be two situations.



You can have homogeneous data types and models. You can have heterogeneous data types and model classes. Here different models are applied on the same data. It avoids specific bias of particular algorithm on a given data set. For example, applying various collaborative filtering engines such as neighborhood based method SPD Bayesian techniques on rating matrixes.

You can have the you can generate the what we were combining R u j values. So this you can generate from all these methods. So you can have neighborhood SPD and Bayesian and finally combine them. But the data source is same. Then heterogeneous data source.

Let $R = [r_{uj}]$ be an $m \times n$ ratings matrix

Let $\hat{R}_1 \dots \hat{R}_q$ be the $m \times n$ *completely specified* ratings matrices

Then, for a set of weights $\alpha_1 \dots \alpha_q$, the weighted hybrid creates a combined prediction matrix $\hat{R} = [\hat{r}_{uj}]$ $\quad \hat{R} = \sum_{i=1}^{q} \alpha_i \hat{R}_i$

in terms of individual entries of the matrix: $\hat{r}_{uj} = \sum_{i=1}^{q} \alpha_i \hat{r}^i_{uj}$

So different classes of models are applied on different data sources. For example, but for a particular situation only same rating matrix you use collaborative filtering. And you for the same recommendation you also have the item specified and you use a content based system using that item detail. So in the first case your data source is the rating matrix. Second case the data source is the item features.

So in this case of course combining this way may not be possible. So there has to be other mechanisms in which you are supposed to predict this R u j values from content based system and combine them together. Now let us see how bagging happen in collaborative filtering. The basic idea in bagging is to reduce the variance component of the error in classification. In bagging Q-training data sets are created with bootstrapped sample sampling.

What is bootstrapped sampling? Sampling with replacement. So in a particular variant of bagging also known as sub-bagging sub-samples of the rows are drawn rather than sampling with replacement. So now there can be various types in this case row wise bootstrapping. In this case the rows of the rating matrix are sampled with replacement to create a new matrix new rating matrix of the same dimension. Row wise sub-sampling the dimension now reduces.

So here you take the rows that are sampled without replacement. Now entry wise bagging in this case the entries in the original rating matrix are sampled with replacement. So it is not row wise it is taking the entire matrix at a time. Second one is also taking the entire matrix at a time. But here in entry wise sub-sampling a fraction of the entries is retained at random from the rating matrix to create the sample training data set.

Now finally all these sub-samples use various data sources which are sampled from the original and make the predictions. So weighted average of this you take as your solution. Now you can also inject randomness in collaborative filtering. When we say by injecting randomness we mean we can we mean we can do it in both neighborhood based model as well as factorization model. In case of neighborhood based model where we take the top k neighbors rather we choose more than top k and select few neighbors k neighbors from within that.

Now average prediction from various components is returned by this approach. In case of matrix factorization, it is already does random initializing while determining the factor matrices. How does it do? Because the algorithm requires the data to be first it is an iterative procedure. So you initialize certain random data and then through iteration you improve. But when you do so the results may vary because the initial it also depends on the initial value.

So therefore what happens to introduce further to inject further randomness you generate these results from different initialization settings. Next comes the switching hybrid. Here the algorithm switches between various recommender systems depending on the current need. The switching mechanism for example, the switching mechanism for cold start issue can be solved by using knowledge based recommender system when few ratings are available or the content based recommender system where the details of the new item is anyway available. So here of course we are going beyond we have not talked about knowledge based system in the next class we will be talking about them.

But this is one additional input additional input. This is not those basic inputs where user matrix item matrix and preference matrix of the rating matrix. But here some additional source will be used. But in collaborative filtering we do not use a content do not use item details. We use item details only in content based system.

So therefore to solve the cold start to at least the item level cold start issues can be solved in collaborative model based in collaborative filtering considering this item based details. We can also use a bucket of models. A fraction 25 to 33 percent is prescribed of the specific entries in the rating matrix are held out and various models are applied to the resulting matrix. The held out entries are then used to evaluate the effectiveness of the model.

This process we have already studied. But here the idea is you use that model which gives you lowest MSE mean squared error or mean absolute error. Then we have cascade hybrids. Cascading we saw in case of cascading we saw in this case cascading giving one input to the other. So here the recommender is allowed to use the recommendation of previous recommendation in algorithm then combine the result to make a final recommendation. So this can happen in two ways successive refinement of recommendation or the process of boosting.

So in this in the first approach a recommender system successively refines the output of the recommender system in the previous iteration. One example could be the first recommender can provide the rough ranking and also eliminate many of the potential items. The second level of recommendation then uses this rough ranking to further refine and break the ties. So you can get this recommendation. How do you get this recommendation? By refining the values of the rating which you obtain from one matrix and maybe you can if you have ties and all while comparing these rate ratings the predicted rating you can have some further consideration using data maybe from some other source and break the time.

Next is boosting in this a sequence of training round is used and weighted with weighted training examples. The weights in each round are modified depending on the performance of the classifier in the previous round. Specifically, the weights on training example with the weights on the training examples with error are increased whereas the weights on the correctly modeled examples are reduced. So this process what happens you can you can get a classifier towards correctly classifying the examples that are that it was not able to properly classify in the previous round.

So there are again feature augmentation hybrids. So as the name indicates you create features by adding features from the previous the output of one recommender system is used as the input feature to the next. For example, here the missing entries in the rating matrix can be estimated by content based system to create a denser matrix. These newly added ratings are preferred as referred as pseudo ratings can be used in the collaborative system for making better prediction. Meta level hybrids so here the model used by one recommender system is used as input to another system. The content based peer groups is used to determine the most similar items of the target user.

So once the peer groups are determined then the weighted average of the rating of the peer groups are used to determine the prediction rating. There are feature combination hybrids where features from different data sources are combined. Here the example could be you can have a augmented rating matrix and add columns for keywords and keywords will come from which source it can come from item description it can come from users reviews and so on. So essentially your rating matrix now which was m cross n now it becomes m cross n plus d where n is the number of items d is the number of keywords. The weights of the keyword items are based on the description of the items assessed bought or rated by the user.

So this is coming from the item description or it can also come from the users reviews the keywords the user himself uses. The traditional neighborhood or matrix factorization approach can apply on this augmented rating matrix. And finally, we have mixed hybrids. Here the main characteristics is they combine the scores from different components in terms of presentation rather than in terms of combining the predicted scores. If you see in a movie recommender system you will be getting the movie on various aspects because you liked this you got this recommendations because similar users saw this movies you got some recommendation some popular movies you got recommendation and so on.

So here is another example a tourism domain may present bundle of recommendations where each bundle contains multiple category of items for example different categories may respond to accommodation leisure activities air ticket and so on. So these are my references I used for this and finally these are the conclusions is hybrid recommender systems are used either to leverage the power of multiple data sources or to improve the performance of existing recommender system within a particular data modality. There are three design paradigms ensemble design monolithic design and mixed system in case of ensemble it can be parallel or sequential but it consists of multiple independent recommender algorithms. But in case of monolithic design you combine and make them a bigger system in case of mixed you think of considering different data sources or different algorithms. With this we conclude this. Thank you.