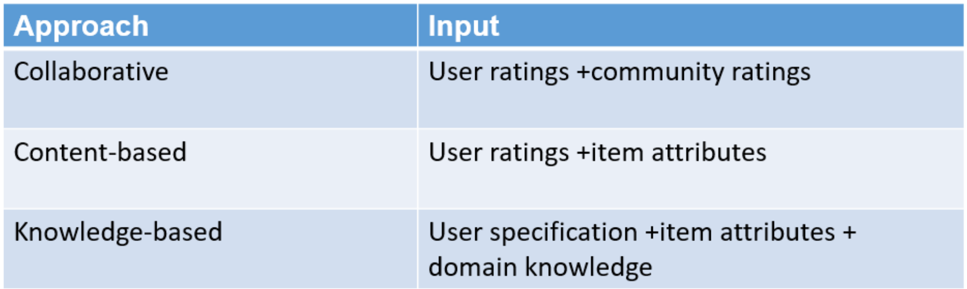
Course Name - Recommender Systems
Professor Name - Prof. Mamata Jenamani
Department Name - Industrial and Systems Engineering
Institute Name - Indian Institute of Technology Kharagpur
Week - 08
Lecture - 37

Lecture 37: Knowledge based recommender systems

Hello everyone! We are going to have the second lecture of this last module and today we are going to discuss about knowledge based recommender system. So to start with, why do we need another source of data for recommendation? So far we have talked about collaborative and content based systems. But both of them suffer from the few problems. To start with, in both the settings the time span plays an important role. As the time changes, the user's requirement keeps changing, the rating pattern keeps changing and the items which he liked at a particular time, the nature of the, the feature of the item which he would like at a later time will also change. Moreover, the product features keeps changing.

When we talk about the product features, if you remember, product features we were also using text features. These text features were derived from two sources, product description and may be from users reviews. So as we add most user reviews, it is likely to change. So in both the cases, time span plays an important role.

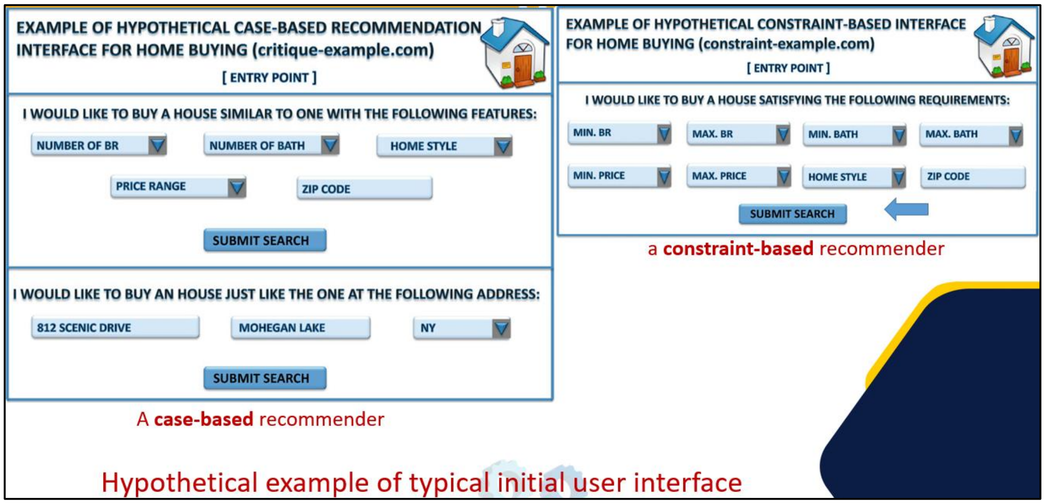
Second is, in both we depend on three inputs. One, two inputs. In case of collaborative filtering, we use the rating matrix and in case of content based, we use the item features and the rating matrix. There is no additional knowledge source about the user's choice. So we cannot explicitly, so in order to explicitly capture customer requirements, we use something called knowledge based system.



Approach	Input
Collaborative	User ratings +community ratings
Content-based	User ratings +item attributes
Knowledge-based	User specification +item attributes + domain knowledge

So in this knowledge based system, the recommended items are based on specific domain knowledge about how certain item features meet user's need and preferences and ultimately how the item is useful for the user. And both these are interactive. Interactive in the sense, they will be presenting certain recommendations, then they will allow the users to change it and keep improving. So here, as I have told you, in collaborative we have user ratings and ratings of other users. A single user's personal rating plus rating of the other users.

That is basically the rating matrix only. In this case, content based, you use user's rating for individual user and item rating. What do you make? You make an item feature and rating. So you make a dataset and you make a user's model. And in the knowledge based system, you use user specification, item attribute plus domain knowledge which you collect otherwise but looking at the behavior of the specification of the last users.



EXAMPLE OF HYPOTHETICAL CASE-BASED RECOMMENDATION INTERFACE FOR HOME BUYING (critique-example.com)

[ENTRY POINT]

I WOULD LIKE TO BUY A HOUSE SIMILAR TO ONE WITH THE FOLLOWING FEATURES:

I WOULD LIKE TO BUY AN HOUSE JUST LIKE THE ONE AT THE FOLLOWING ADDRESS:

A **case-based** recommender

EXAMPLE OF HYPOTHETICAL CONSTRAINT-BASED INTERFACE FOR HOME BUYING (constraint-example.com)

[ENTRY POINT]

I WOULD LIKE TO BUY A HOUSE SATISFYING THE FOLLOWING REQUIREMENTS:

a **constraint-based** recommender

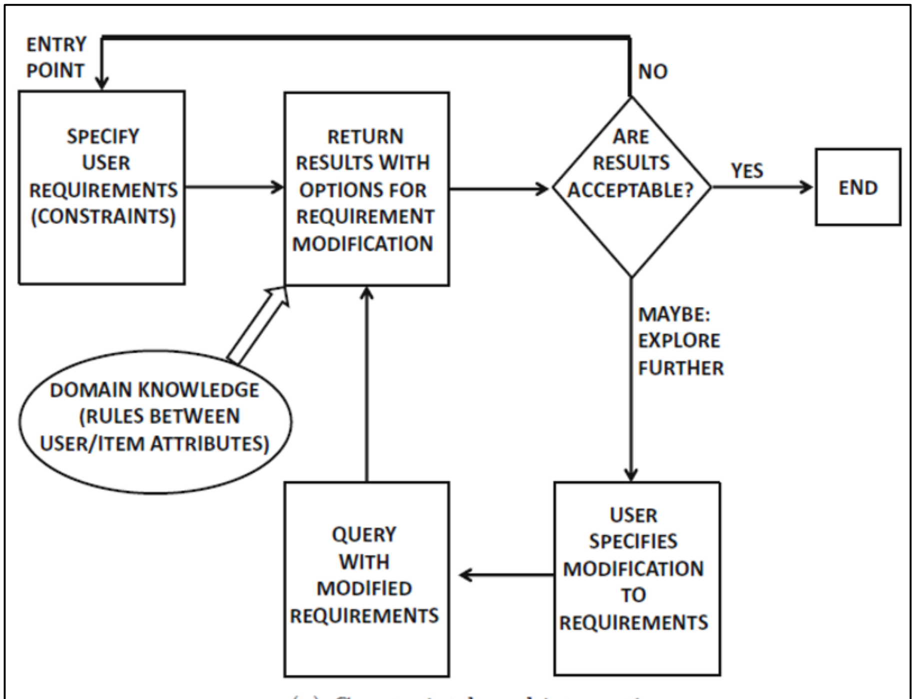
Hypothetical example of typical initial user interface

So this is an additional source of data you now use. Now these are the typical questions that are answered in a knowledge based recommender system. What kind of domain knowledge can be represented in a knowledge base? What mechanism can be used to select and rank the items based on user's characteristics? How do we acquire the user profile in domain in which no purchase history is available? How can we take the customer's explicit preferences into account? Now which interaction pattern can be used in the interactive while interacting with the user? In which dimensions can we personalize the dialogue to maximize the precision of the preferential elicitation process? So there are two broad type of knowledge based recommender system. Constraint based and case based. In case of constraint based system, users need are represented in the form of few constraints and they try to satisfy those constraints with products.

Because you are recommending products, so the products which satisfy those constraints you recommend. And when you do so, you have to use some specifically defined rule set and this rule set make your knowledge base. The items that fulfill this recommendation are suggested. The second one is your case based recommender system. So here instead of using the rule, you use something called the cases.

So what are the cases? Similar kind of suggestions. So here the similarities make the knowledge base. Now here it tries to retrieve the items that are similar to the specification. In both constraints based and case based approaches, conversation takes place. User first specifies the requirement.

System tries to identify the solution. If the user is if the solution is found, the user is allowed to make for the refinement if he is not happy and if and he continues. In both cases the user the system provides an opportunity for the user to change the specified requirements. But the way the change happen varies in both cases. So as I told you in the first case, in case based it uses similarity matrix and in the second constraint based it uses certain decision rules.

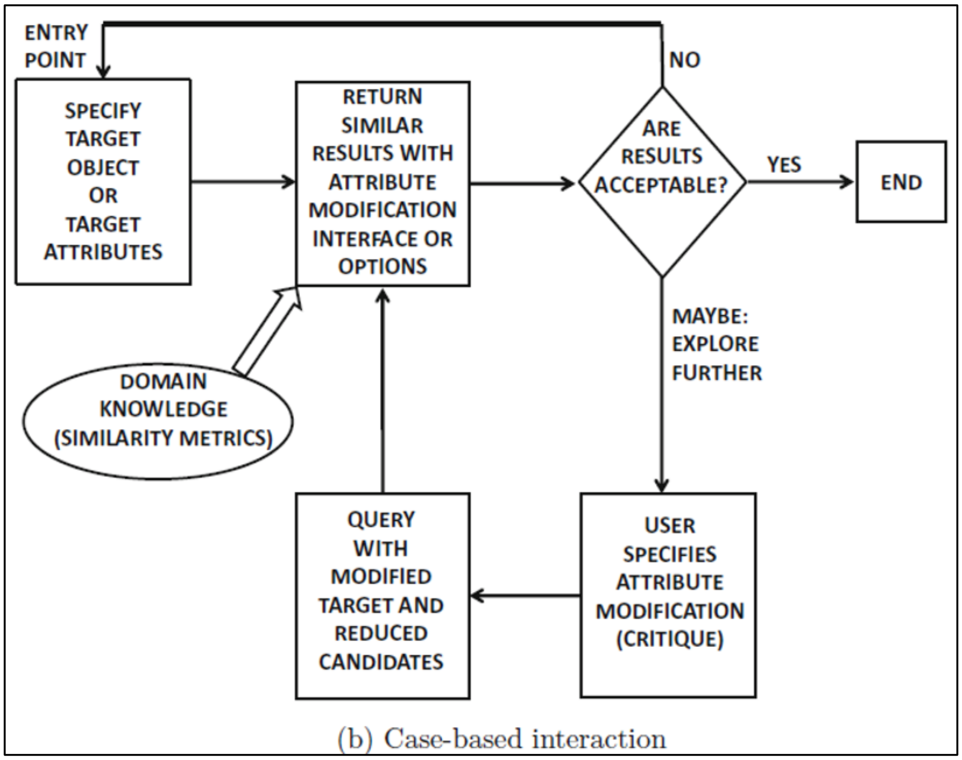

ENTRY POINT
NO
SPECIFY USER REQUIREMENTS (CONSTRAINTS)
RETURN RESULTS WITH OPTIONS FOR REQUIREMENT MODIFICATION
ARE RESULTS ACCEPTABLE?
YES
END
MAYBE: EXPLORE FURTHER
DOMAIN KNOWLEDGE (RULES BETWEEN USER/ITEM ATTRIBUTES)
QUERY WITH MODIFIED REQUIREMENTS
USER SPECIFIES MODIFICATION TO REQUIREMENTS

(a) Constraint-based interaction

This is one hypothetical example taken from Charva Groval book. In fact, for all these additional recommender systems we are mostly referring Charva Groval recommended system book. So here as you can see in case of here you have a constraint based and here you have a case based system. Here the user is asked to provide certain criteria which comes in forms of constraints. What is the minimum number of bedrooms, maximum number of bedrooms? This is some kind of home buying, home buying.

Minimum bedrooms, maximum bedrooms, minimum bathroom, maximum bath, minimum price, maximum price, style, zip code for the area. So here you ask for the similarity. Look here you are asking for the requirement, here you are asking for the features so that you can match. This is one way of doing it. Another is it can show you various houses and one that is of particular house type you can also ask.

So that the features of this house can be used to provide you further recommendation in the maybe in this particular area. So this is again the whole idea that we discussed is presented in the form of a flowchart. So user starts specifying the requirement, maybe one screen just now you saw one of such screens. Then gives the requirement, looks at the result. If he is happy, it ends.


ENTRY POINT
NO
SPECIFY TARGET OBJECT OR TARGET ATTRIBUTES
RETURN SIMILAR RESULTS WITH ATTRIBUTE MODIFICATION INTERFACE OR OPTIONS
ARE RESULTS ACCEPTABLE?
YES
END
DOMAIN KNOWLEDGE (SIMILARITY METRICS)
MAYBE: EXPLORE FURTHER
QUERY WITH MODIFIED TARGET AND REDUCED CANDIDATES
USER SPECIFIES ATTRIBUTE MODIFICATION (CRITIQUE)

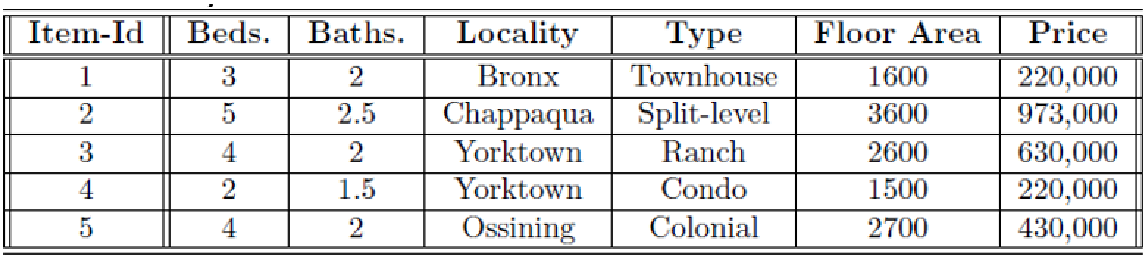
(b) Case-based interaction

If you would like to completely refresh, you come back and start. But if you are partially shown some results and you would like to partially modify your, you have to partially modify your requirement. You modify once more query is made, results is compared, user is shown. If he is happy, he goes out. Otherwise he may choose to continue in this loop.

Then in case of case based, same things happen. If you look at this, same thing happens, same you loop. One thing I forgot to tell you, this initial set of rules make the rule base and this rule base can also be updated for looking at the user's behavior. But this is the domain knowledge which goes in. Similarly domain knowledge here comes in form of similarity matrix.

That is the input to this comparison process. And as explained, this here also you repeat the search process. You iterate through the searching procedure. So more on constraint based system. In this case, user explicitly specified the requirement or the constraints, lower and upper bound on certain item attributes and so on.

The domain specific rules are used to match the user requirement or attributes to item attributes. These rules represent domain specific knowledge used by the system. You saw there was a rule base. Such rules could take the form of domain specific constraints and the item attributes. Here for example, somebody is trying to buy cars.



Item-Id	Beds.	Baths.	Locality	Type	Floor Area	Price
1	3	2	Bronx	Townhouse	1600	220,000
2	5	2.5	Chappaqua	Split-level	3600	973,000
3	4	2	Yorktown	Ranch	2600	630,000
4	2	1.5	Yorktown	Condo	1500	220,000
5	4	2	Ossining	Colonial	2700	430,000

Cars before the year 1970 do not have cruise control. So this is a rule. So if somebody is searching for the cars before that some vintage or something, he or she cannot give this as one of the constraint. So such rules will be pre-existing. Then the system may create rules relating user attributes to item attributes.

So older investors do not invest in ultra high risk products. This can be one of the observations from the past data which can be converted to rules. User attributes may also be specified in the search process. Depending on the number of and types of returned results, as we saw user has to loop. So some initial then user or sometimes it may so happen that there is no result.

So user will be actually relaxing the constraint and give and if large number of results are coming up, then user will be now limiting the search process, limiting the bounds. So knowledge representation. So items features can be represented in the forms of knowledge. With different items what are the features together which come together can make a rule. Now customer requirements like the typical customer attributes can also be used to form rules.

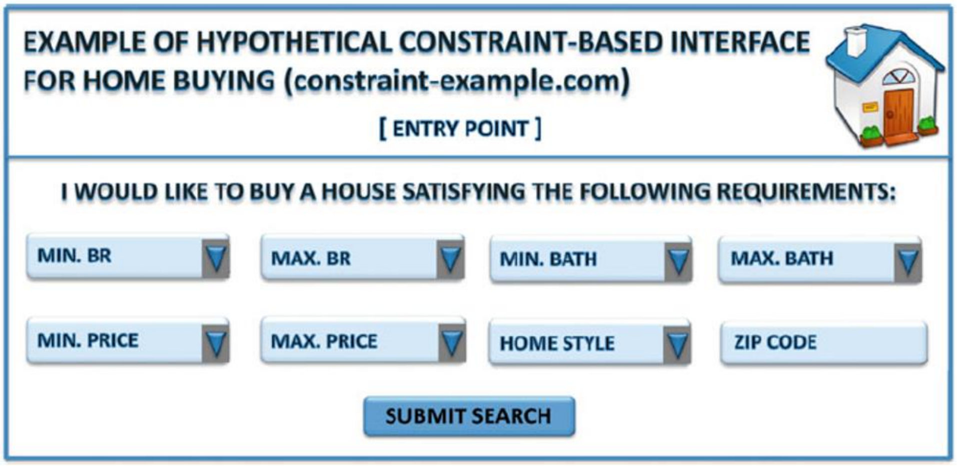
So these are some of the typical attributes. Now let's say you are talking about marital status, then you know that people single people may not, will not typically require very large number of bedrooms. So if somebody is single, then you can make a rule like this. If somebody is single, the minimum bedroom, if the marital status is single, the minimum bedroom should be less than equal to 5. If family size is greater than 5, minimum number of bedrooms is this.

So these are some of the rules which you can decide a prior. Now while returning relevant results, you can solve one constraint satisfaction problem by viewing each item in the catalogue as they as constraint on attributes and expressing the catalogue in disjunctive normal form. This expression is then combined with the rules in the knowledge base to determine whether the mutually consistent region of product space exists. If they do not, what happens? Customer is not given any choice, so he has to by himself relax the rules and come back. And if large number of items are recommended, then he has to increase the constraint level and come back.

So these could be some of the rules which are triggered and got the items back. Interaction approach. In this, an interactive interface is shown to the user and a ranked list of matching item is presented and they will have certain values and there will be some weights. So you make some utility function for the user. So as you keep on refining this data, this value of this utility function changes and at each instance high utility items will be shown to the user.

$$\boxed{U(\overline{V}) = \sum_{j=1}^{d} w_j \cdot f_j(v_j)}$$

So these are some of the situations you may encounter. Handling the options are not selected. Basically you fire a query and get the output from the database. The query and the product database with available constraint you get and use the default from past data or from some search statistics. You can also handle certain, you also have to have procedures to handling empty sets and have provisions to repair the constraints.



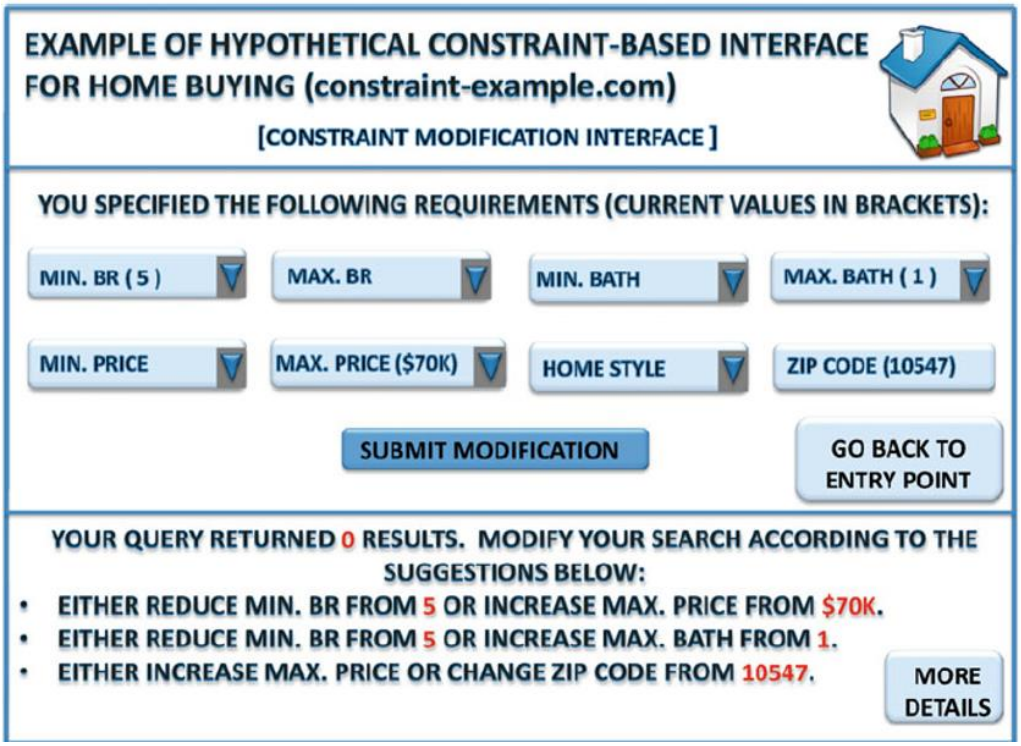
EXAMPLE OF HYPOTHETICAL CONSTRAINT-BASED INTERFACE FOR HOME BUYING (constraint-example.com)

[ENTRY POINT]

I WOULD LIKE TO BUY A HOUSE SATISFYING THE FOLLOWING REQUIREMENTS:

This example we saw last time. So somebody is trying to buy the house and these are shown to define the constraint on the selection. Suppose now based on this you get these options. So what is the option? You got zero choice. So you are given certain hints that your bedroom should be like this, then your bath should be like this, preference should change to certain other area which may be nearer to whatever you specified and based on that you change. So it may so happen that you get very large number.

So if you get a very large number of results, you may add put additional constraints and for that suggestions are given. So based on these suggestions you submit your modifications and get the result. Or if you are happy you can go back to the entry point. Second is your case based recommender system. In case of case based system as you know instead of rules the similarities make the knowledge base.



EXAMPLE OF HYPOTHETICAL CONSTRAINT-BASED INTERFACE FOR HOME BUYING (constraint-example.com)

[CONSTRAINT MODIFICATION INTERFACE]

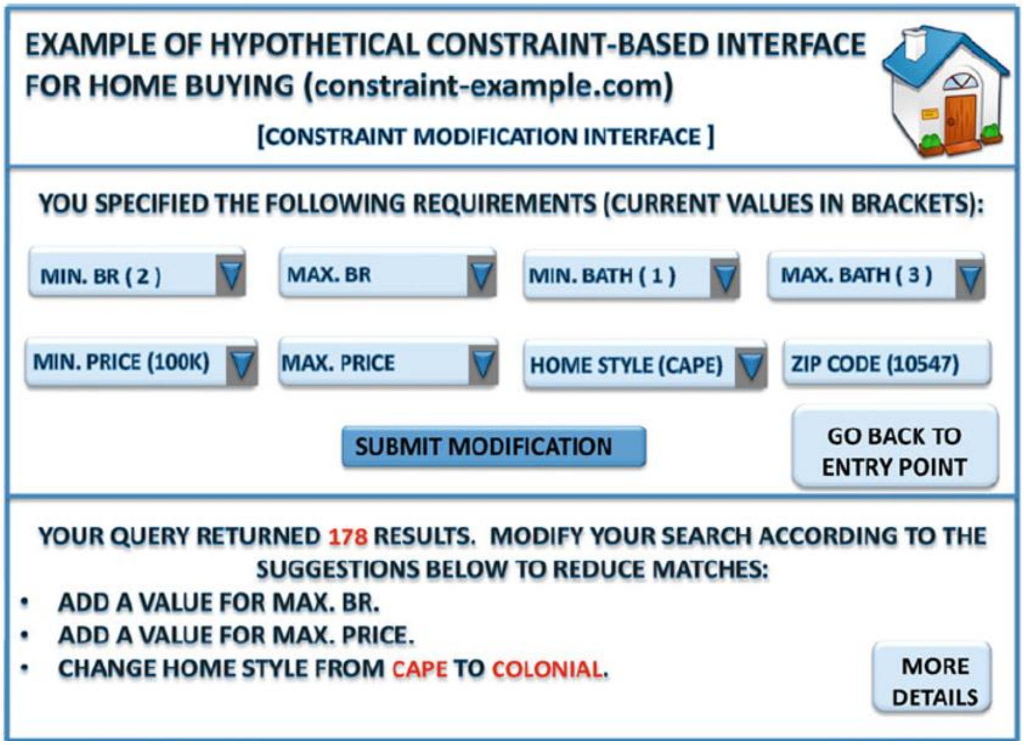
YOU SPECIFIED THE FOLLOWING REQUIREMENTS (CURRENT VALUES IN BRACKETS):

MIN. BR (5) | MAX. BR | MIN. BATH | MAX. BATH (1)

MIN. PRICE | MAX. PRICE ($70K) | HOME STYLE | ZIP CODE (10547)

YOUR QUERY RETURNED 0 RESULTS. MODIFY YOUR SEARCH ACCORDING TO THE SUGGESTIONS BELOW:

- EITHER REDUCE MIN. BR FROM 5 OR INCREASE MAX. PRICE FROM $70K.
- EITHER REDUCE MIN. BR FROM 5 OR INCREASE MAX. BATH FROM 1.
- EITHER INCREASE MAX. PRICE OR CHANGE ZIP CODE FROM 10547.



EXAMPLE OF HYPOTHETICAL CONSTRAINT-BASED INTERFACE FOR HOME BUYING (constraint-example.com)

[CONSTRAINT MODIFICATION INTERFACE]

YOU SPECIFIED THE FOLLOWING REQUIREMENTS (CURRENT VALUES IN BRACKETS):

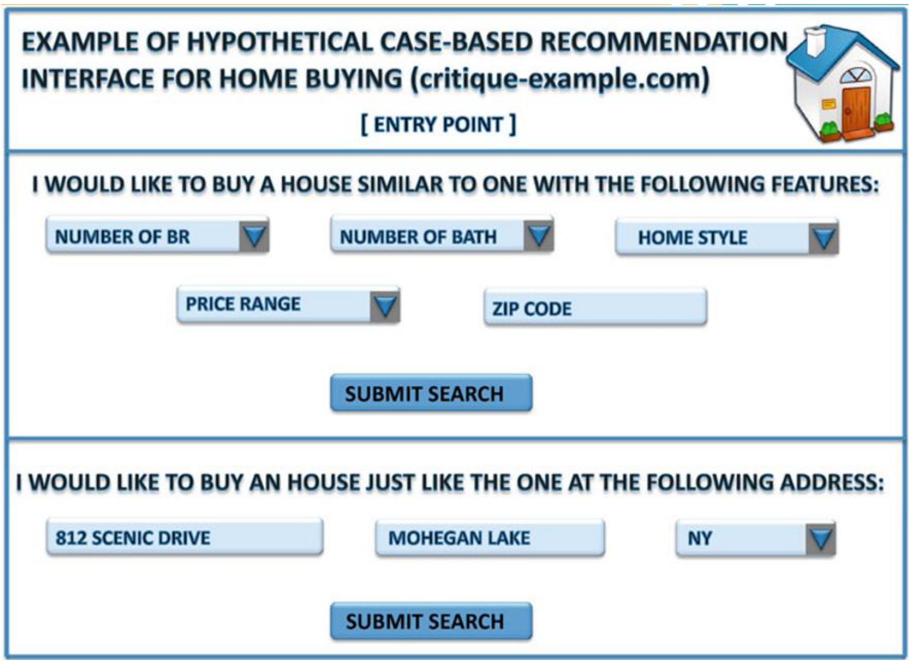
MIN. BR (2) | MAX. BR | MIN. BATH (1) | MAX. BATH (3)

MIN. PRICE (100K) | MAX. PRICE | HOME STYLE (CAPE) | ZIP CODE (10547)

YOUR QUERY RETURNED 178 RESULTS. MODIFY YOUR SEARCH ACCORDING TO THE SUGGESTIONS BELOW TO REDUCE MATCHES:

- ADD A VALUE FOR MAX. BR.
- ADD A VALUE FOR MAX. PRICE.
- CHANGE HOME STYLE FROM CAPE TO COLONIAL.

So here the similarity matrix are used to retrieve the examples that are comparable to the specific target or case. Here the constraints are not enforced in terms of rules. The results are refined. The repeated modification of user's requirement is done through a process called critiquing.



EXAMPLE OF HYPOTHETICAL CASE-BASED RECOMMENDATION INTERFACE FOR HOME BUYING (critique-example.com)

[ENTRY POINT]

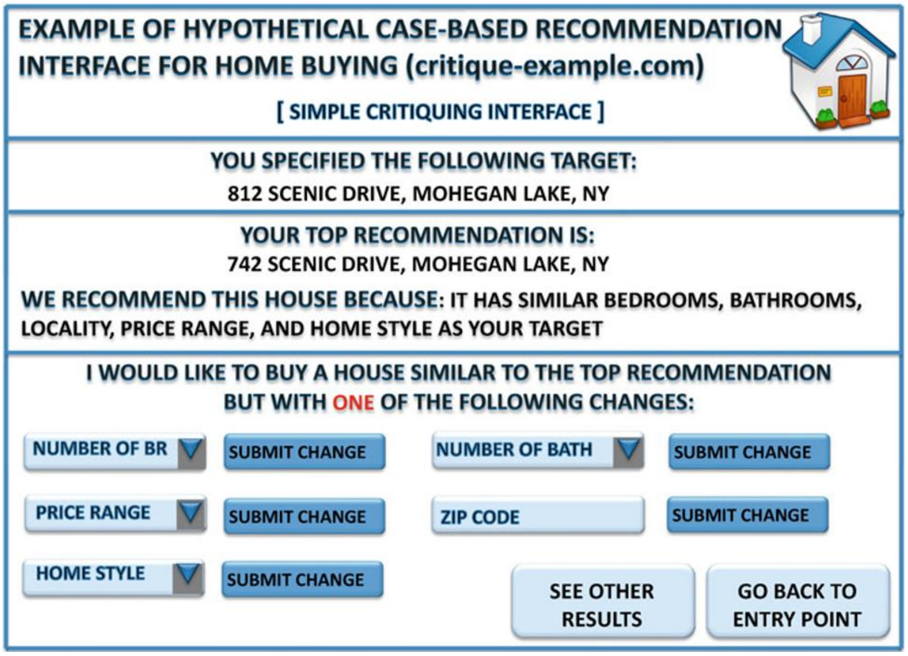
I WOULD LIKE TO BUY A HOUSE SIMILAR TO ONE WITH THE FOLLOWING FEATURES:

I WOULD LIKE TO BUY AN HOUSE JUST LIKE THE ONE AT THE FOLLOWING ADDRESS:

812 SCENIC DRIVE

MOHEGAN LAKE

NY



EXAMPLE OF HYPOTHETICAL CASE-BASED RECOMMENDATION INTERFACE FOR HOME BUYING (critique-example.com)

[SIMPLE CRITIQUING INTERFACE]

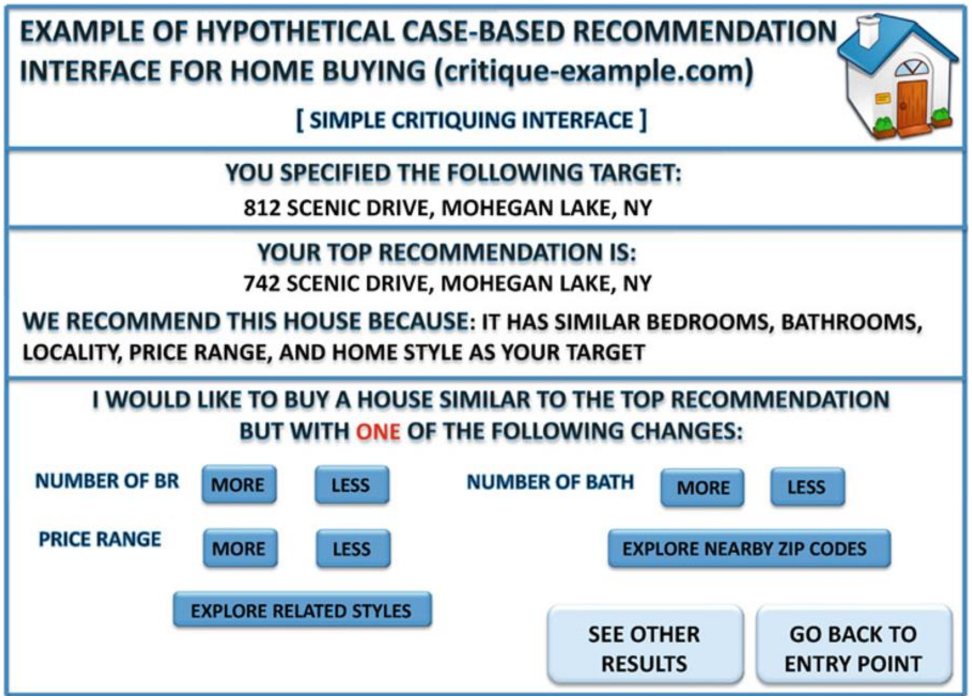
YOU SPECIFIED THE FOLLOWING TARGET:
812 SCENIC DRIVE, MOHEGAN LAKE, NY

YOUR TOP RECOMMENDATION IS:
742 SCENIC DRIVE, MOHEGAN LAKE, NY

WE RECOMMEND THIS HOUSE BECAUSE: IT HAS SIMILAR BEDROOMS, BATHROOMS, LOCALITY, PRICE RANGE, AND HOME STYLE AS YOUR TARGET

I WOULD LIKE TO BUY A HOUSE SIMILAR TO THE TOP RECOMMENDATION BUT WITH ONE OF THE FOLLOWING CHANGES:

So this critiquing process is of three types. First is the single critiquing where one change for attribute for one feature is allowed at a time. It is time taking because user has to do it one attribute at a time. In case of compound critiquing user gets the facility to specify multiple modifications at a time. So then you also have something called dynamic critiquing. It is similar to compound critiquing but more relevant possibilities are represented using data mining approach.



EXAMPLE OF HYPOTHETICAL CASE-BASED RECOMMENDATION INTERFACE FOR HOME BUYING (critique-example.com)

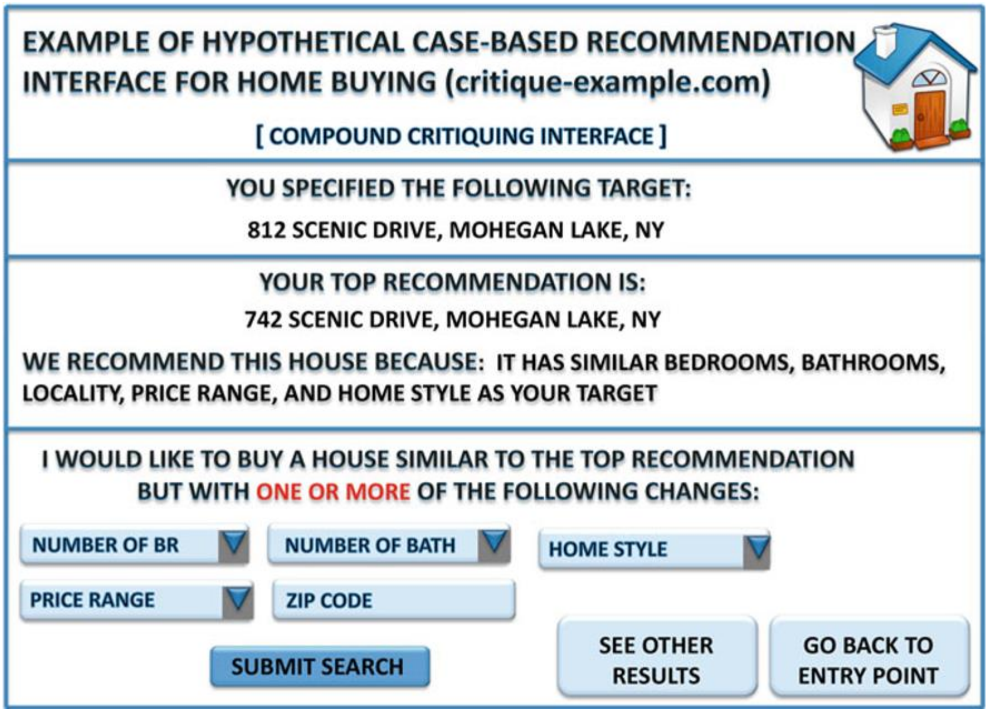
[SIMPLE CRITIQUING INTERFACE]

YOU SPECIFIED THE FOLLOWING TARGET:
812 SCENIC DRIVE, MOHEGAN LAKE, NY

YOUR TOP RECOMMENDATION IS:
742 SCENIC DRIVE, MOHEGAN LAKE, NY

WE RECOMMEND THIS HOUSE BECAUSE: IT HAS SIMILAR BEDROOMS, BATHROOMS, LOCALITY, PRICE RANGE, AND HOME STYLE AS YOUR TARGET

I WOULD LIKE TO BUY A HOUSE SIMILAR TO THE TOP RECOMMENDATION BUT WITH ONE OF THE FOLLOWING CHANGES:



EXAMPLE OF HYPOTHETICAL CASE-BASED RECOMMENDATION INTERFACE FOR HOME BUYING (critique-example.com)

[COMPOUND CRITIQUING INTERFACE]

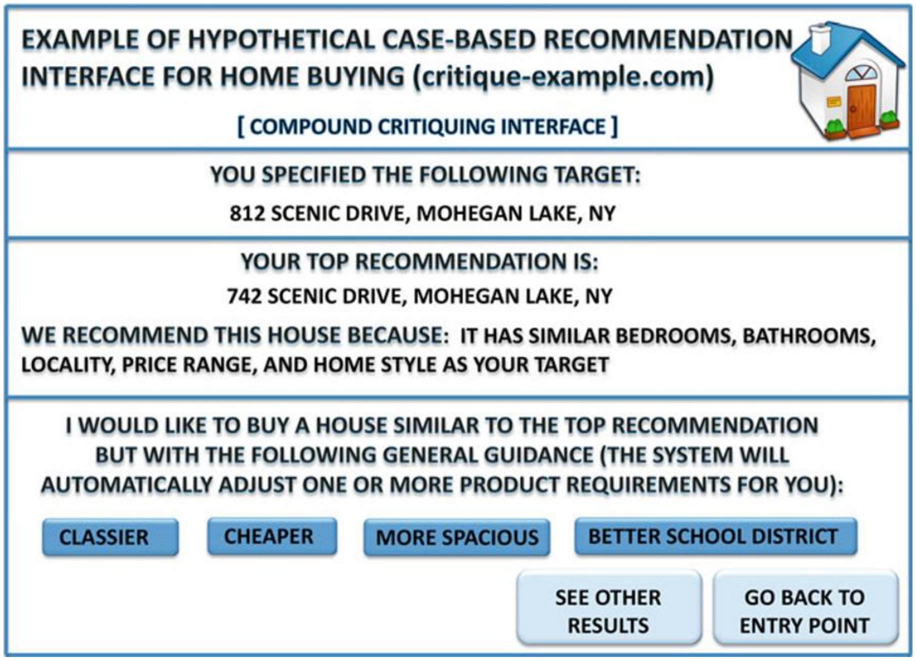
YOU SPECIFIED THE FOLLOWING TARGET:
812 SCENIC DRIVE, MOHEGAN LAKE, NY

YOUR TOP RECOMMENDATION IS:
742 SCENIC DRIVE, MOHEGAN LAKE, NY

WE RECOMMEND THIS HOUSE BECAUSE: IT HAS SIMILAR BEDROOMS, BATHROOMS, LOCALITY, PRICE RANGE, AND HOME STYLE AS YOUR TARGET

I WOULD LIKE TO BUY A HOUSE SIMILAR TO THE TOP RECOMMENDATION BUT WITH ONE OR MORE OF THE FOLLOWING CHANGES:

For example, some frequent pattern analysis etc. can be used for this purpose. This is again one example figure where he is supposed to specify his inputs. Then based on this you get some recommendations and some reason is also given, explanation is also given why this recommendation is done. So if you like take it otherwise you try modifying. In case you submit like this you have to make necessary changes and modify and iterate.



EXAMPLE OF HYPOTHETICAL CASE-BASED RECOMMENDATION INTERFACE FOR HOME BUYING (critique-example.com)

[COMPOUND CRITIQUING INTERFACE]

YOU SPECIFIED THE FOLLOWING TARGET:

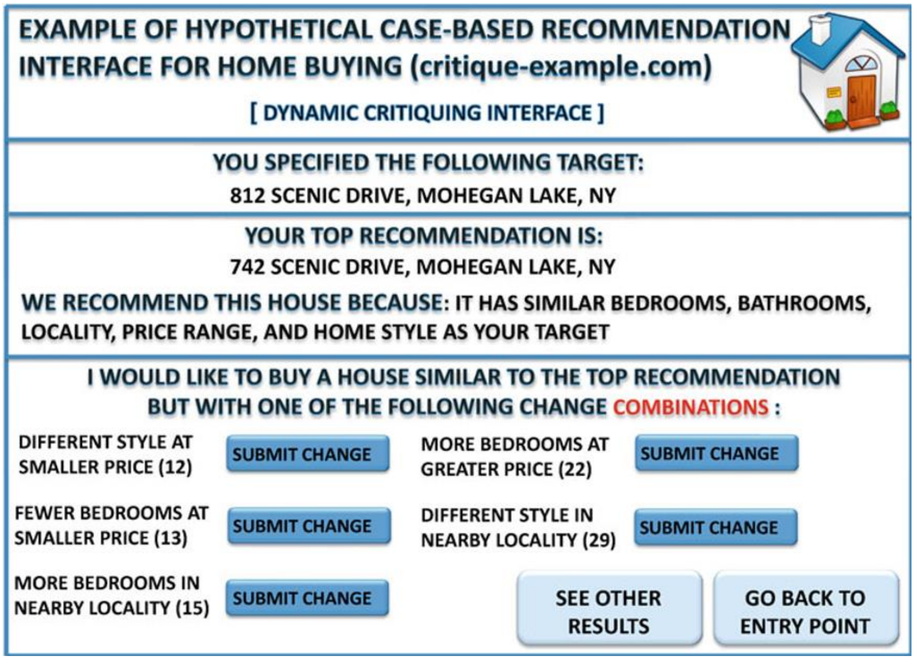
812 SCENIC DRIVE, MOHEGAN LAKE, NY

YOUR TOP RECOMMENDATION IS:

742 SCENIC DRIVE, MOHEGAN LAKE, NY

WE RECOMMEND THIS HOUSE BECAUSE: IT HAS SIMILAR BEDROOMS, BATHROOMS, LOCALITY, PRICE RANGE, AND HOME STYLE AS YOUR TARGET

I WOULD LIKE TO BUY A HOUSE SIMILAR TO THE TOP RECOMMENDATION BUT WITH THE FOLLOWING GENERAL GUIDANCE (THE SYSTEM WILL AUTOMATICALLY ADJUST ONE OR MORE PRODUCT REQUIREMENTS FOR YOU):



EXAMPLE OF HYPOTHETICAL CASE-BASED RECOMMENDATION INTERFACE FOR HOME BUYING (critique-example.com)

[DYNAMIC CRITIQUING INTERFACE]

YOU SPECIFIED THE FOLLOWING TARGET:

812 SCENIC DRIVE, MOHEGAN LAKE, NY

YOUR TOP RECOMMENDATION IS:

742 SCENIC DRIVE, MOHEGAN LAKE, NY

WE RECOMMEND THIS HOUSE BECAUSE: IT HAS SIMILAR BEDROOMS, BATHROOMS, LOCALITY, PRICE RANGE, AND HOME STYLE AS YOUR TARGET

I WOULD LIKE TO BUY A HOUSE SIMILAR TO THE TOP RECOMMENDATION BUT WITH ONE OF THE FOLLOWING CHANGE COMBINATIONS :

DIFFERENT STYLE AT SMALLER PRICE (12)

MORE BEDROOMS AT GREATER PRICE (22)

FEWER BEDROOMS AT SMALLER PRICE (13)

DIFFERENT STYLE IN NEARBY LOCALITY (29)

MORE BEDROOMS IN NEARBY LOCALITY (15)

So every time you will be getting the explanation as well. So here you have got some other suggestion and you can provide the guidance to the system through your suggestion. Asking for the changes, modify and submit. So these are very I mean I have used these references specifically Charva Agrawal's all the examples have been taken from there and these are the conclusions. Knowledge based systems recommend items based on specific domain.

Knowledge about how certain item features meet users need and preference is of utmost importance here. Then constraint based recommender systems. It is of two types constraint based and case based. In case of constraint based you have the rules in you have the domain knowledge in the form of rules knowledge base and here the domain knowledge is in case of case based it is the similarities.

You solve a constraint satisfaction problem. You solve it by comparing the solution. In both the cases the items the user goes through certain iterative process to provide the suggestion. So with this we finish this knowledge based system. Thank you.