Recommendation System for How to Get to Know Someone

Zeyao Yang
PhD student in Indiana University Bloomington
zeyyang@indiana.edu

Abstract

With increase of work's difficulty, people have to cooperate with others. In this situation, people want to obtain a comfortable way to know person they want to, so need for such recommendation system has been increasing. Thus, I am trying to build this kind of system. In my project, I built a recommendation system based on CBR, and revised some part of ordinary CBR system to make my system become smarter. The experiment results demonstrate that my system is able to achieve expected goal. Moreover, my project also provides creative idea to apply CBR into broader fields.

Problem

With development of network, there are a lot of data online. Therefore, researchers are interested in how to use these data. One of popular issues is to use data to do recommendation. From example, recommendation system in Facebook helps users to find friends that they want to know or already know before; product recommendation system in Amazon recommend products that users want to buy in the future.

In my project, I am trying to build a new recommendation system that helps people to find a comfortable way to know the person that they want to know. Here, comfortable way means a way by which people can leave good impression. For instance, I am a researcher in Chicago University, and I am interested in CBR system. There is one day, I read paper from professor David Leak whose research interest is CBR, and think the paper is so amazing. Thus, I want to cooperate with him. However, as in common sense, Professor Leak possible is not willing to cooperate with researcher he never knows before. In this situation, I want to find a comfortable way to introduce myself and give a good impression. It will be great if a system can do such kind of recommendation. For example, the system finds that Professor Leak will attend a conference next week in Chicago University, and draws a conclusion that he is willing to know other researcher in academic meeting according to personality information. And then, the system will suggest me to go to the meeting and introduce myself. The final goal of my project is to build this kind of recommendation system.

Although there are currently many recommendation systems, none of them focus on the issue that my project is addressing. On the one hand, it is difficult to collect enough data to provide this recommendation. On the other hand, because practical situation is really complicated, it is difficult to build one single algorithm to reach precise prediction. However, with increase of work difficulty, people's need to find suitable partner will increase in the future. Thus, the issue that my project focuses on will become popular and attract more attention in next ten years.

Now, most of recommendation systems predict by analyzing the graph, which is abstracted from knowledge domain. However, in my project, I am trying to build a recommendation system based on CBR, which will provide creative way to solve this problem. Moreover, if we abstract the problem that my project is addressing, we find that the problem is similar with issue that is to predict attribute of new link in a graph. Thus, my project is able to provide creative idea for graph analysis and will be applied in more fields in the future.

Related Work

The core of my project is to do recommendation based on CBR system, so the related works mainly focus on two aspects.

Recommendation System

With the development of technology, people have closer contact with others, and it is possible that they could not remember every person they know. Moreover, they also have a lot of choice for product that they want to buy, but it is difficult to find which one is the best one for them. In this situation, recommendation system becomes necessary [2]. Currently, two issues are very popular for recommendation system:

As in common sense, a lot of data is available online, but people do not known how to use them. Thus, the re-

search in recommendation model for large databases is really attractive [3]. Related work not only includes how to collect data online as suitable format to use, but it also incudes how to utilize collected data to do recommendation and prediction.

Moreover, many researchers now try to use graph theory to solve recommendation and prediction problem [5]. In there models, they abstract collected data as graph. Nodes in graph represent data entity, and links between nodes represent relationship between entities. The analysis for graph mainly focuses on two aspects: node information and topology information. Because models that focus on topology information do not rely on specific knowledge, they achieve bigger success.

CBR

CBR is area under Artificial Intelligence, and it is based on the idea that machine will be as smart as human beings if it is able to summarize previous experience like human [6]. The most famous application of CBR is IBM Watson.

In the last few years, case-based reasoning has developed from an isolated research area to a field that attracts widespread interest.

There are many methods in CBR. First of all, cases may be concrete experiences, or a set of generalized cases. And, cases not only can be stored as separate knowledge units, but it also can be spitted up into subunits. Secondly, CBR systems can index cases by a prefixed or hierarchical index structure. Thirdly, solution can be obtained by directly applying a previous case, or modified current solution according to differences between cases. Furthermore, adaptation of solutions can be supported by a complex model from domain knowledge, or by syntactic similarity only. At last, CBR methods can be totally automatic, or they can interact with the users. Each aspect of above has been done a lot of research in past few years.

Summary

In my project, I am trying to solve a problem in recommendation field. The problem is similar to predict unrelated attributes of new link in a graph. However, I did not simply use ordinary graph theory methods. I creatively apply CBR to do recommendation. In my system, current methods for recommendation can also be used if we modify them as rules. Moreover, I also revise some steps in CBR to make system smarter with increase of data. In sum, I use revised CBR to solve an important issue in recommendation field.

Motivation

Although many recommendation systems haves been applied in practice, none of them can help people find comfortable ways to get to know the person that they want to know. Thus, I am trying to build such kind of system to satisfy people's need. Moreover, instead of relying on general knowledge of a specific domain, or only use changeless algorithms to do recommendation, CBR recommendation system is able to utilize previous data. More cases system addresses, it becomes smarter and smarter.

Method

I analyze my recommendation system based on CBR. The core in CBR system is to solve new problems that are called new case by retrieving previous cases and adapting them to solve new problem.

At fact, there are steps in CBR system: 1) there should be a lot of cases in database; 2) new case is inputted into system; 3) system indexes whole database and tries to find all similar cases, called candidate cases; 4) some cases are totally same with new case, and some cases are only a little bit similar with new case. Therefore, system will do adaptation that means to combine and modify solutions of all similar cases to product most possible solution for the new case; 5) users or system evaluate the new solution and revise it; 6) if the solution is goo enough, the system will store the new case and its solution.

My project not only follows the above basic CBR system, but it also adds several creative ideas and modifies several parts to adapt the specific recommendation.

Choose Case

In my project, I take the relationship between two people as case. For example, if the system known that Jianqi got to know Max by presentation, the system will store Jianqi-Presentation-Max as a case. It is unnecessary to take more complex relationship as case, because system has to spend more space to store case and more time to store and release case. More importantly, more complex case will not mean more precise prediction. Thus, to achieve the initial goal more concisely, I make decision to take a single relationship as a case.

Build Rules

The next step is to build rules to address new cases. According to the six degrees of separation theory, people are able to get to know the person that they want to know by at most six people. Therefore, it is safe to say that path is good information that can be used to predict. From above analysis, I build 5 rules in my system.

- Rule0 will be used when the relationship already exists in the system.
- Rule1 will be used when user and person that user wants to know have common neighbors.

- Rule2 will be used when the length of shortest path between user and the person that user wants to know is two.
- Rule3 will be used when the length of shortest path between user and the person that user wants to know is three.
- Rule4 will be used when all of above rules cannot address new cases.

Designers of system can input any rule that they want. In my system, I mainly focus on revised path information, which is enough for recommending comfortable way to know other classmates in B552 class. However, in the future, when people want to apply the system idea into more complex situation, they can add more rules.

Unlock Rules

This part is the most creative revision for CBR system. At the beginning of my system, there is not any case in database, and all rules are disabled except rule4. With increase of case, if cases satisfy requirement to unlock a rule, this rule will be enabled, and will be used in the future.

This idea has two positive influences:

On the one hand, system will not use rules that have not been verified by practical data, so it is more likely to provide precise recommendation for users. Moreover, with collecting more data and enabling more rules, system will become smarter.

On the other hand, designers of system can input any rule that they want, but these rules will keep disabled until cases activate them. Therefore, designers do not have to think about if these rules are good enough, because system will choose rules automatically according to data. Designers just input all rules that they can come up with, and let system to choose suitable rules. Thus, this idea not only helps system become smarter for users, but it also makes system become smarter for designers.

Adaptation

When system addresses new case, each rule will compute his similarity with new case. And each rule also has its own accuracy. In this step, system will compare each role's similarity and accuracy, and choose the best recommendation for users.

Evaluation and Add Case

My system will let users to make decision if recommendation is right. After above process, system will return a way, and asks user "Do you accept this recommendation?" If users accept it, this relationship will be added to database. Otherwise, system does nothing.

When new case enters system, system will try to activate new rules and re-compute accuracy of each rule.

Summary

The method is built on idea of CBR system, and adds some creative ideas in each step. Comparing with regular CBR system, my system needs to have suitable rules, so the most difficult part in my project is to build rules that fit recommendation in the class.

If we take the problem to larger viewpoint, we find that the core problem is to predict link attribute, which is an interesting problem in graph theory. Applying CBR to solve this problem is creative. More importantly, this method does not require all attributes are related.

At last, the method will have broader application in the future. Because the case in the system is relationship, system can be used in many fields as long as it focuses on relationship. Furthermore, the relationship is not only constrained into human relationship, but it also includes broader definition such as relationship between produce and user. To achieve broader application, designer need only to input rules that fit new environment.

Experiment and Result

We assume that database is empty at the beginning. In the following experiment, I will gradually add cases into database to test if new suitable new cases will enable more rules and reach more precise recommendation.

1) At the beginning, only is rule4 enabled. When Zeyao wants to know Max, system has following result:

```
rule4 is enabled!
Please input your first name:
Zeyao
Please input  first name of the person you want to know
Max
This situation is similar with rule4
The similarity: 0.8
The Rule4's accuracy: 0.25
Overall adaptation: 0.2
According to above analysis
We recommend: Zeyao should know Max by TeamProject
Do you accept this recommendation?(y/n)
y
Do you want to continue?(y/n)
y
1
Zeyao TeamProject Max
```

2) After first step, there is one case in database. Repeat first step between Max and Ratish, and system gives us right recommendation.

```
Please input your first name:
Max
Please input  first name of the person you want to know
Ratish
This situation is similar with rule4
The similarity: 0.8
The Rule4's accuracy: 1
Overall adaptation: 0.8
According to above analysis
We recommend: Max should know Ratish by TeamProject
Do you accept this recommendation?(y/n)
y
Do you want to continue?(y/n)
y
2
Zeyao TeamProject Max
Max TeamProject Ratish
```

3) After second step, there are two cases in databases. When Ratish wants to know Andrew, the system gives wrong recommendation as following:

```
Please input your first name:
Ratish
Please input  first name of the person you want to know
Andrew
This situation is similar with rule4
The similarity: 0.8
The Rule4's accuracy: 1
Overall adaptation: 0.8
According to above analysis
We recommend: Ratish should know Andrew by TeamProject
Do you accept this recommendation?(y/n)
n
Do you want to continue?(y/n)
y
2
Zeyao TeamProject Max
Max TeamProject Ratish
```

4) In last step, because user gives negative feedback, system will try other recommendation next time:

```
Please input your first name:
Ratish
Please input  first name of the person you want to know
Andrew
This situation is similar with rule4
The similarity: 0.8
The Rule4's accuracy: 0.333333
Overall adaptation: 0.266667
According to above analysis
We recommend: Ratish should know Andrew by Presentation
Do you accept this recommendation?(y/n)
y
Do you want to continue?(y/n)
y
3
Zeyao TeamProject Max
Max TeamProject Ratish
Ratish Presentation Andrew
```

5) Similar, when Andrew wants to know Jordan, the system has following results:

```
Please input your first name:
Andrew
Please input  first name of the person you want to know
Jordan
This situation is similar with rule4
The similarity: 0.8
The Rule4's accuracy: 0.5
Overall adaptation: 0.4
According to above analysis
We recommend: Andrew should know Jordan by Presentation
Do you accept this recommendation?(y/n)
y
Do you want to continue?(y/n)
y
4
Zeyao TeamProject Max
Max TeamProject Ratish
Ratish Presentation Andrew
Andrew Presentation Jordan
```

6) When Jordan wants to know Zeyao and accepts recommendation, rule3 is enabled! Zeyao and Jordan are connected after adding the new case, and path between them is four if we remove direct relationship between them, which satisfies rule3's requirement.

```
Please input your first name:
Jordan
Please input  first name of the person you want to know
Zeyao
This situation is similar with rule4
The similarity: 0.8
The Rule4's accuracy: 0.6
Overall adaptation: 0.48
According to above analysis
We recommend: Jordan should know Zeyao by Presentation
Do you accept this recommendation?(y/n)
y
Do you want to continue?(y/n)
y
5
Zeyao TeamProject Max
Max TeamProject Ratish
Ratish Presentation Andrew
Andrew Presentation Jordan
Jordan Presentation Zeyao
rule3 is enabled!
Rule3's Accuracy: 1
```

7) When Andrew wants to know Jianqi, system will still use rule4 because this case is more similar with rule4.

```
Please input your first name:
Andrew
Please input  first name of the person you want to know
Jianqi
This situation is similar with rule4
The similarity: 0.8
The Rule4's accuracy: 0.666667
Overall adaptation: 0.533333
According to above analysis
We recommend: Andrew should know Jianqi by Presentation
Do you accept this recommendation?(y/n)
y
Do you want to continue?(y/n)
y
6
Zeyao TeamProject Max
Max TeamProject Ratish
Ratish Presentation Andrew
Andrew Presentation Jordan
Jordan Presentation Zeyao
Andrew Presentation Jianqi
Rule3's Accuracy: 0.714286
```

8) Now, when Zeyao wants to know Jianqi, the new case is more similar with rule3. Thus, system will use rule3 to recommend. More importantly, with adding such a new case, rule2 is enabled, because current data satisfies rule2's requirement.

```
Please input your first name:
Zeyao
Please input  first name of the person you want to know
Jianqi
This situation is similar with rule3
The similarity: 0.8
The Rule3's accuracy: 0.714286
Overall adaptation: 0.571429
According to above analysis
We recommend: Zeyao should know Jianqi by Presentation
Do you accept this recommendation?(y/n)
y
Do you want to continue?(y/n)
y
7
Zeyao TeamProject Max
Max TeamProject Ratish
Ratish Presentation Andrew
Andrew Presentation Jordan
Jordan Presentation Zeyao
Andrew Presentation Jianqi
Zeyao Presentation Jianqi
rule2 is enabled!
Rule2's Accuracy: 0.307692
Rule3's Accuracy: 1
```

9) Now, when Max wants to know Jordan, the new case is more similar with rule2. Thus, system will use rule2 to recommend. Similarly, after adding such a new case, rule1 is enabled, because current data satisfies rule1's requirement.

```
Please input your first name:
Max
Please input  first name of the person you want to know
Jordan
This situation is similar with rule2
The similarity: 0.8
The Rule2's accuracy: 0.307692
Overall adaptation: 0.246154
According to above analysis
We recommend: Max should know Jordan by TeamProject
Do you accept this recommendation?(y/n)
y
Do you want to continue?(y/n)
y
8
Zeyao TeamProject Max
Max TeamProject Ratish
Ratish Presentation Andrew
Andrew Presentation Jordan
Jordan Presentation Zeyao
Andrew Presentation Jianqi
Zeyao Presentation Jianqi
Max TeamProject Jordan
rule1 is enabled!
Rule1's Accuracy: 0.214286
Rule2's Accuracy: 0.533333
Rule3's Accuracy: 1
```

10) Under current database, when Jordan wants to know Jianqi, system will use rule1 because rule1 is able to provide better adaptation.

```
Please input your first name:
Jordan
Please input  first name of the person you want to know
Jianqi
This situation is similar with rule1
The similarity: 1
The Rule1's accuracy: 0.214286
Overall adaptation: 0.214286
According to above analysis
We recommend: Jordan should know Jianqi by Presentation
Do you accept this recommendation?(y/n)
y
Do you want to continue?(y/n)
n
9
Zeyao TeamProject Max
Max TeamProject Ratish
Ratish Presentation Andrew
Andrew Presentation Jordan
Jordan Presentation Zeyao
Andrew Presentation Jianqi
Zeyao Presentation Jianqi
Max TeamProject Jordan
Jordan Presentation Jianqi
```

Above results exactly match with real data collected in class. From experiment result, we can find that rules are enabled with increase of database. Moreover, when more rules are enabled, system can use them to do better recommendation.

Conclusion

From above experimental result, we can draw a conclusion that the system performs really well.

My project has several creative things. On the one hand, I take relationship as case and use CBR idea to deal with this problem, which will broaden application of CBR. On the other hand, system will enable different rules according to actual situation, and this is a new idea to make the system smarter. Thirdly, when analyzing, every rule has his similarity and accuracy, it will make the adaptation more accurate.

However, my system also has its own limitation. First of all, five rules are not enough. The practical situation is more complex, if we still want to do good recommendation we have to excavate more information from different aspects, which means more rules. For example, we can use path counter model. Secondly, because of limitation of data and time, system focuses on B552 class. If we want to implement practical system, we have to consider more possible recommendation results rather than only TeamProject, Presentation, Discussion and FinalProject.

From this project, I learned how to implement a real CBR system.

Acknowledgments

I thank Professor David Leak for giving me such an opportunity to do the real CBR system. I thank Mingze for his commend and suggestion on this project. I also thank our classmates for doing survey.

References

[1] Althoff, Klaus-Dieter. "Case-based reasoning." Handbook on Software Engineering and Knowledge Engineering 1 (2001): 549-587.

[2] Ricci, Francesco, Lior Rokach, and Bracha Shapira. Introduction to recommender systems handbook. Springer US, 2011.

[3] Jafarkarimi, Hosein, Alex Tze Hiang Sim, and Robab Saadatdoost. "A naive recommendation model for large databases." International Journal of Information and Education Technology 2.3 (2012): 216-219.

[4] Resnick, Paul, and Hal R. Varian. "Recommender systems." Communications of the ACM 40.3 (1997): 56-58.

[5] Mirza, Batul J., Benjamin J. Keller, and Naren Ramakrishnan. "Studying recommendation algorithms by graph analysis." Journal of Intelligent Information Systems 20.2 (2003): 131-160.

[6] Aamodt, Agnar, and Enric Plaza. "Case-based reasoning: Foundational issues, methodological variations, and system approaches." AI communications 7.1 (1994): 39-59.