Final Project Proposal

Predict decisions on Weibo with Factorization Machine

Kyuhyun Cho(20133677) Seunghwan Baek(20133331)

Joongyoun Lee(20115494) Hyeonjun Jo(20133693)

1. PROBLEM

How can we get to know if a person choose something or not? Basically we can ask them directly. However, how about guessing the decision? To guess is a different problem. We need to refer to any information which is related to the decision. Sometimes we refer to some information even not related at all superficially. At this point, we think that perhaps A.Is could do the 'guessing'. They could find the related information, look into it, and compile statistics on it. Then, they could find some rules and finally make a decision. This is what we are doing here. There are amazingly countless data on webs especially Weibo. We are going to make an A.I using that data, and train it. Finally the A.I is going to make a decision if a user will choose to follow an item(a person, a group, a product anything) recommended. There are so many data can be used. Unless we do not use all the data, we need to choose some data to train A.I. Therefore this choosing task is supposed to be most challenging. They can be examined alone, or sometimes be combined to make a more accurate decision.

2. DATA

You will see below what we need to cope with. To deal with them we are going to look into through all data, and pick some data only we need. After picking up, the data will be denoised. Maybe we will remove some dummy users and duplicate data. We are also concerning doing dimensional reduction for a 'diet' of the data.

3. METHOD

Before mentioning about how to do, there is the list of what we are going to focus on. The first is temporal dynamics. This means trends of items. If an item is recommend to users when the item is very trendy, the users' following the item has relatively high probability. So we are going to deal with this temporal dynamics. The second is a users' active period. Supposed a user is really actively doing Weibo at the moment. And there is a recommendation one item to that user. There would be a stronger chance to accept the item compared to when the user is not so active. The next is the implicit feedback. This is not just the target user's interest. We are going to focus more on the users who are similar to the target user. If there are some common items among the users, we could guess that the target user also could be interested in the items. Lastly, the sequential information is also really important. This can be called pattern. We had better know this information to guess. This has vary data. One of them is the time gap between one item recommended and another item recommended. Whit this gap, we can get to know how long users spend their time some items.

Basically we are about to use Factorization Machines which are a new model that combines the advantages of Support Vector Machines with factorization models. These are a general predictor working with any real valued feature vector. Especially these model interactions between variables using factorized parameters. Due to the modelling FMs can estimate interactions even in problems with huge sparsity.[1] And the following issues is how to do these FMs. The answer is the libFM. This is an implementation of FM. For libFM we need a binary data format though, there is a converter form the standard text format to the binary data format.[2] We will see through all about this soon.

Lastly, we have been looking for other useful and creative things such as methods, logics, or features which can be help for this problem. We could not help using FMs because of its performance. However we reckon it would be worth to implement something our own.

4. IMPLEMENT

We mainly use C++, because C++ is easy to implement libFM. And we probably use Python partially.

5. REFERENCES

[1] Steffen Rendle, "Factorization Machines," in the Institute of Scientific and Industrial Research Osaka University, Japan

[2] Steeffen Rendle, "Factorization Machines with libFM," in University of Konstanz