Journal of Advances in Mathematics Vol 16 (2019) ISSN: 2347-1921 https://cirworld.com/index.php/jam

DOI: https://doi.org/10.24297/jam.v16i0.8212

The computational approach for recommendation system based on tagging data

Kateryna Nesvit

Department of Applied Mathematics, Karazin Kharkiv National University,

Maidan Svobody, 4, office 6-32, Kharkiv 61022, Ukraine.

e-mail: nesvit@karazin.ua

Abstract

Recommendation approaches like a platform for learning algorithm. We can use some predicted values to put them in the data pipeline for learning. There is a hard nuance of how to calculate the similarity measure when we have a small number of actions at all, it's not a new user or item to use cold start methods, we just have not enough quantity to say it may be interpreted like regularity. The frequency of tags what we would have from users will have a huge impact to predict his future taste. The article describes created a computational approach using as explicit and as implicit feedbacks from users and evaluates tags by Jaccard distance to resolve this issue. To compare results with existed numerical methods there is a comparison table that shows the high quality of the proposed approach.

Keywords: recommendation systems, numerical methods, algorithms.

Introduction

There are a few tricky words what sounds simple it's "similarity" and "recommendation". What is "similarity"? Which parameters could determine it more closely? What is the difference between "prediction" and "recommendation" because we often heard these words.

We have a lot of choices every day around us and it becomes really annoying to waste our time just trying to find relevant books, papers, news, people, films and other things. There is a good book about choice [1] it's "The paradox of choice: Why more is less" by Barry Schwartz. It's a useful book to better understand what the choice is. Recommendation system helps us in this complicated way because we also wish to get the right information immediately without any asked questions. It means we should read the user's thoughts. How can we do that? How we can help the user take the right decision? The key idea of recommendation is simple just chose some similar items to users and recommend it but this supposes erroneously because there are a lot of nuances for each case in that field where we suppose to use it.

The recommendation system is a young field of research. There is something above it's machine learning approach that covers prediction. But much more appealing is the idea of developing automatic approaches which can optimize the performance of the learning algorithm to compute and discover something new from that data we have. We never know what we are going to get, but we can try to guess.

Data is everything and all feedbacks from users or items are growing day by day. If we collect all of this information it will be batch learning or in the opposite case if we use the only current data it may consider like an adaptive approach to prediction. There is a thin slit between the time what we should take from the past to now to be able to follow nowadays tastes of users. It often happens when we faced with recommendation what keeps our older interest. You have bought it! It's annoying and it makes the user disappointed. Because we don't follow his thought enough as he wishes. Wouldn't it be great to know more about our users without having to ask them?

There is a key to start using some learning algorithm when we don't have enough raw data to train models to get the accuracy we would want.

Interesting approach to recommendation using tags was proposed in [2] to extract association rules from folksonomies and use them to recommend supplementary resources associated with the tags involved in these rules. It even better to find tags which frequently appear together, it will more personalize the behavior of our user. As noticed in [3] relationship information among people, tags, and items are collected and aggregated across different sources and results show a significantly better interest ratio for the tag-based recommender than for the people-based recommender.

There is a lot of fields where we could apply tags techniques and their logic to build a strong system to help people stay with personalized information. It is matching music with the user's location [4], finding image tags are related to the image's visual content [5], social bookmarking systems and their emergent information structures [6].

It's so intuitive to see that [7] tagging has emerged as a powerful mechanism that enables users to find, organize, and understand online entities. Algorithms combining tags with recommenders may deliver both the automation inherent in recommenders and the flexibility and conceptual comprehensibility inherent in tagging systems. Could we take useful knowledge from evaluating tags that related to that item? Namely this core idea lays in [7] where was shown an efficient algorithm of how to predict users' preferences for items based on their inferred preferences for tags. Hybrid technique in a similar case was shown using tag-based collaborative filtering based on the semantic distance among tags assigned by different users to improve the effectiveness of neighbor searching [8]. But what if we haven't responded as a clicks from tags or we haven't not enough time to calculate it because it's hard to track of them all. The only question is what we can do in this case.

This paper describes an approach when we have some user's feedback to items, we have tags what he puts to items and other interaction between users and items. Consider a nuance when we cannot use collaborative filtering approach because we have not enough users to be close to the target user for prediction. And supplemented it is might not a good way to use association rule approach because it was happened not enough times or just still not enough to be more personalize if we already have enough tags from users.

1 Mathematical model

New approaches and some hybrid methods appear with such frequency that the evaluations of their usefulness are at least slightly out of date by the time are published. But at the same time, these new technologies make our life easier for many people.

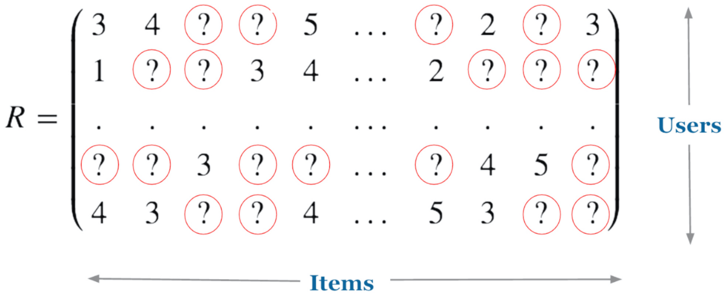

Figure 1: Initial matrix with users and rated items

The matrix into Fig. 1 is a problem statement of the recommendation system. The more questions you are going to ask yourself with this matrix, you will be surprised it becomes complicated each time. Our prediction should be really accurate to care about the behavior of users in the whole website or application. Because even a small number of action can involve huge impact. In the opposite case, even the efficient algorithm is not worth noticing. We should predict what someone is going to like. To be able to analyze the feedbacks and how our algorithm is working, obviously we need a tool for reliability, for example, root means square error Eq. 1.

$$RMSE = \sqrt{\frac{1}{N_p}(r_{i,j} - \widehat{r_{i,j}})^2} \tag{1}$$

One of the traditional approaches to predict value is collaborative filtering that uses a weighted rate of users or items which are similar to the target position. What means "similar items"? And what users are dissimilarity to each other. How to calculate the similarity between persons and items. There are a lot of ways to do this, a lot of approaches of choice. Which is the best for your field, for your needs? There are some nuances as a difference between the length of vectors, not enough similarity value to consider as close to target user / item, not enough the elements in the vector to be compared. To cover these issues we choose Jaccard similarity measure Eq. 2 to evaluate compared words.

$$Sim(x, y) = \frac{\mid x \cap y \mid}{\mid x \mid + \mid y \mid - \mid x \cap y \mid} \tag{2}$$

Table 1 shows us basic notations to better explaining the concepts of the proposed approach by this paper.

It's often the case when user and items have some tags, why don't we use it? Every user has his own taste profile it's like of everything that you like and our goal is analyzing this profile accurately.

Table 1: Summary of Notations

Symbol	**Description**
R	input Martix
nU	number of Users
nI	number of Items
nfb	number of feedback from User to Item
$r_{i,j}$	known Rating from *i-th* User to *j-th* Item
$\widehat{r_{i,j}}$	predicted Rating from i-th User to *j-th* Item
N_p	number of predicted Rating
PSM	Percentage of Singularity Matrix
$PnoPM$	Percentage of no Predicted Matrix
$RMSE$	Root Mean Square Error
Tag_i	*i-th* Tag
fvI_q	fuzzy value of *q-th* Item
$Sim(i,j)$	similarity value between *i-th* row and *j-th* column of similarity matrix
$idTgU_i$	ID of *i-th* Tag of some User
$idTgI_j$	ID of *j-th* Tag of some Item
$fTgU_i$	frequency of *i-th* Tag of some User
$fTgI_j$	frequency of *j-th* Tag of some Item
$nSTg_{i,j}$	number of similarity Tags between *i-th* User and *j-th* Item
nTg	set of how many tags the User or Item has
$tSTag$	threshold point of similarity Tags
$mR_{i,j}$	mean Rate thought *i-th* user or *j-th* item choosing the maximum set

2 Computational algorithms

Deep investigation into the theory exists to make a sense only if it's necessary by main computation goal. The more complicated algorithm the harder to get useful data to take knowledge and interpret them to be the right way. For this reason, the creation of concept data model and logic model are the important first step to build a mathematical model for any real problem to play with.

The key idea to create this approach is to take a more influent piece of information between words and to give substantial weight to the recommendation. We are going to use a defuzzification technique Eq. 3 to calculate the fuzzy value of the scaled item.

$$fvI_q = \frac{\sum\limits_k Sim(k,q)r_{k,q}}{\sum\limits_k Sim(k,q)} \tag{3}$$

First, we compare two words by Algorithm 1. To use Jaccard similarity Eq. 2 in Algorithm 1 we should find a common letters. There is a question of how we can do this if every word could be with repeated letters and they have own positions. We can lose a semantic and sense at all. That's why we divide a set between cases if we have repeated letters into each word or no. If yes it's simple just calculate the

$$R = \begin{array}{c} \\ \\ \end{array}\begin{array}{ccccc} & & \text{q-th Item} & & \\ 4 & ? & ? & 4 & 2 \\ 4 & 5 & 4 & 3 & 5 \\ 1 & 4 & ? & 1 & 1 \\ 5 & ? & 4 & 5 & 1 \\ ? & 2 & 3 & 1 & 3 \end{array} \qquad Sim = \begin{array}{ccccc} & & \text{q-th Item} & & \\ 0.8 & 0.7 & 0.7 & 0.9 & 1 \\ ? & 0.6 & 1 & 1 & 0.6 \\ 1 & 1 & 0.9 & 0.9 & 0.7 \\ 0.8 & 0.8 & 0.7 & 0.9 & 0.9 \\ ? & ? & 0.9 & 0.8 & ? \end{array}$$

Figure 2: Sample of Data for prediction by Tags

intersection between each set of letters and stay care about at what position they are. Another case is if we have even one repeated letter from one of the considered word. For doing this we should use a common letters function represented by Algorithm 2.

Algorithm 1: Computation of Similarity Tags

```
Input: $Tag_i, Tag_k$
Output: $Sim(i,k)$ is value of Similarity between Tags
$a = size(Tag_i)$, $b = size(Tag_k)$
if $Tag_i = Tag_k$ then
    c=a
else
    if have not any repeated letters into each word then
        $coulped = intersect(Tag_i, Tag_k)$ find a set of intersection
        $c \leftarrow$ calculate how many letters are on true order position
    else
        $c \leftarrow commontLetters(Tag_i, Tag_k)$
    end
end
$Sim(i,k) = \frac{c}{a+b-c}$
```

UniqueSet function is using to find a set of letters for each word without repeated letters and save their original letter's position. The *commonLetters* function is using to find a way to calculate what part we need to add to the variable of c to be considered as a similar letter. If we find a letter which is on the same position and have a totally the same frequency of repeated into words it becomes simple just add the maximum size of that letter. But the question is what if we have different frequency of repeated letters and their position are some similar and some dissimilar. For this case, there is using a few sets. One from them is a size of the intersection of sets considered letter into both sets and another set is a maximum of sizes of sets with considers letter. Doing this with each tag we will find a similarity between all the tags we have.

To build recommendation and choice some item to recommend we need a relationship between users and items (see Algorithm 3). We already have a table with the similarity between tags. And we know that each

Algorithm 2: Function of commonLetters

```
Input: $Tag_i, Tag_k$
Output: $c$ is a value of common letters into Tags
calculate $UniqueSet(Tag_i)$ and $UniqueSet(Tag_k)$
$c = 0$
for all $i^{th}$ letters into less Unique Value Set do
    if exist $i^{th}$ letter into bigger Unique Value Set then
        $temp \leftarrow$ vector of positions of $i^{th}$ letter
        if a Set of positions of both word are equal then
            $c = c + K$
            K is size of Set with $i^{th}$ letter
        else
            $c = c + \frac{P}{Q}$
            P is size of intersection of Sets $i^{th}$ letter into both Sets
            Q is max of sizes of Sets with $i^{th}$ letter
        end
    end
end
```

Algorithm 3: Computation of Similarity Users and Items

```
Input: $idTgU_i, fTgU_i, idTgI_j, fTgI_j$ are vectors with IDs and their frequency
Output: $Sim(i, j)$ is value of Similarity between $i^{th}$ User and $j^{th}$ Item
$tsvUI_{i,j} = [], tfUI_{i,j} = []$ are empty vectors of Tag's Similarity and Frequency
for all $p^{th}$ and $q^{th}$ Tags from $i^{th}$ User and $j^{th}$ Item do
    if $Sim\left(idTgU_i(p), idTgI_j(q)\right) > threshold\,point$ then
        $tsvUI_{i,j} \leftarrow Sim\left(idTgU_i(p), idTgI_j(q)\right)$
        $tfUI_{i,j} \leftarrow 0.5\left(fTgU_i(p) + fTgI_j(q)\right)$
    end
end
for all $s^{th}$ enough Similarity Tags between $i^{th}$ User and $j^{th}$ Item do
    calculate weighted Tag Sum
    $wtsUI_{i,j} \leftarrow tsvUI_{i,j} tfUI_{i,j}$
    calculate Normalization Tag Coefficient
    $ntcUI_{i,j} \leftarrow tfUI_{i,j}$
end
$Sim(i, j) = \frac{wtsUI_{i,j}}{ntcUI_{i,j}}$
```

user and item could have tags. The main direction here is calculated into one number how user and item similarity using the id and their frequency. First, we will check if there is enough similar value to consider those tags as a similar and then we add this value and mean of value their frequency to the vector between current user-item. Doing this for all tags what we have from user and item we have two vectors to calculate one value of similarity. We can do this just using weighted sum and normalize it.

May it looks simple but to build it with even a simulation Data Set it requires patience and power of observation. User and Item interaction will be shown as a number from 0 to 1, how they are close to each other. Getting a prediction value for *i-th* User for *j-th* Item by Eq. 4

$$\widehat{r_{i,j}} = Sim_{i,j} fvI_j \tag{4}$$

In case when we have no fvI_j, this value could be replaced to $mR_{i,j}$.

3 Numerical results

It's interesting to see an algorithmic perspective from decrease RMSE because it's trying to bring how is valuable proposed approach could be. Fig. 3 shows us a numerical result of PSM for the current matrix, RMSE between known and predicted values, PnoPM of the predicted matrix where we considered from 10 to 100 users and items. PSM for this case is from 70 to 97 percentage. On these plots CF is green points, AR is blue points and Tags is red points.

Enough value of tags it's not an easy word, we should probably know how much is it in number to correct switch between approaches. For example, here we use 56 different tags for simulation data for this computational experiment and in this case for that data ”enough quality of tags” it's about 9 to 12 tags with 1 or 2 frequencies.

Table 2 shows a case where a number of feedback bigger. The more feedbacks we will have the sparsity of the matrix is reduced and it means it's not a problem to worry about. Even CF approach will be good and AR will be even faster with really big data for this issue.

Table 2: Evaluation of influent parameters for matrix with 100 users and items

Number feedbacks	2 - 3			5 - 7			9 - 12		
Mean Values	CF	AR	TG	CF	AR	TG	CF	AR	TG
PSM	97.49			93.99			89.45		
RMSE	2.01	2.46	1.81	1.83	2.88	1.62	1.54	2.91	1.55
PnoPM	90.43	31.32	22.40	19.54	6.09	19.7	0.02	1.99	22.31

Time for calculation of AR approach is quicky in 2 times compared to CF approach but using TG takes a time, compared to CF what could take less than 1 sec. TG take 3 or 4 sec. Numerical results were calculated using Matlab.

Conclusions

The proposed approach can be used as an additional part of the algorithm recommendation system in a case with a low value of feedbacks while we have enough quantity of tags.

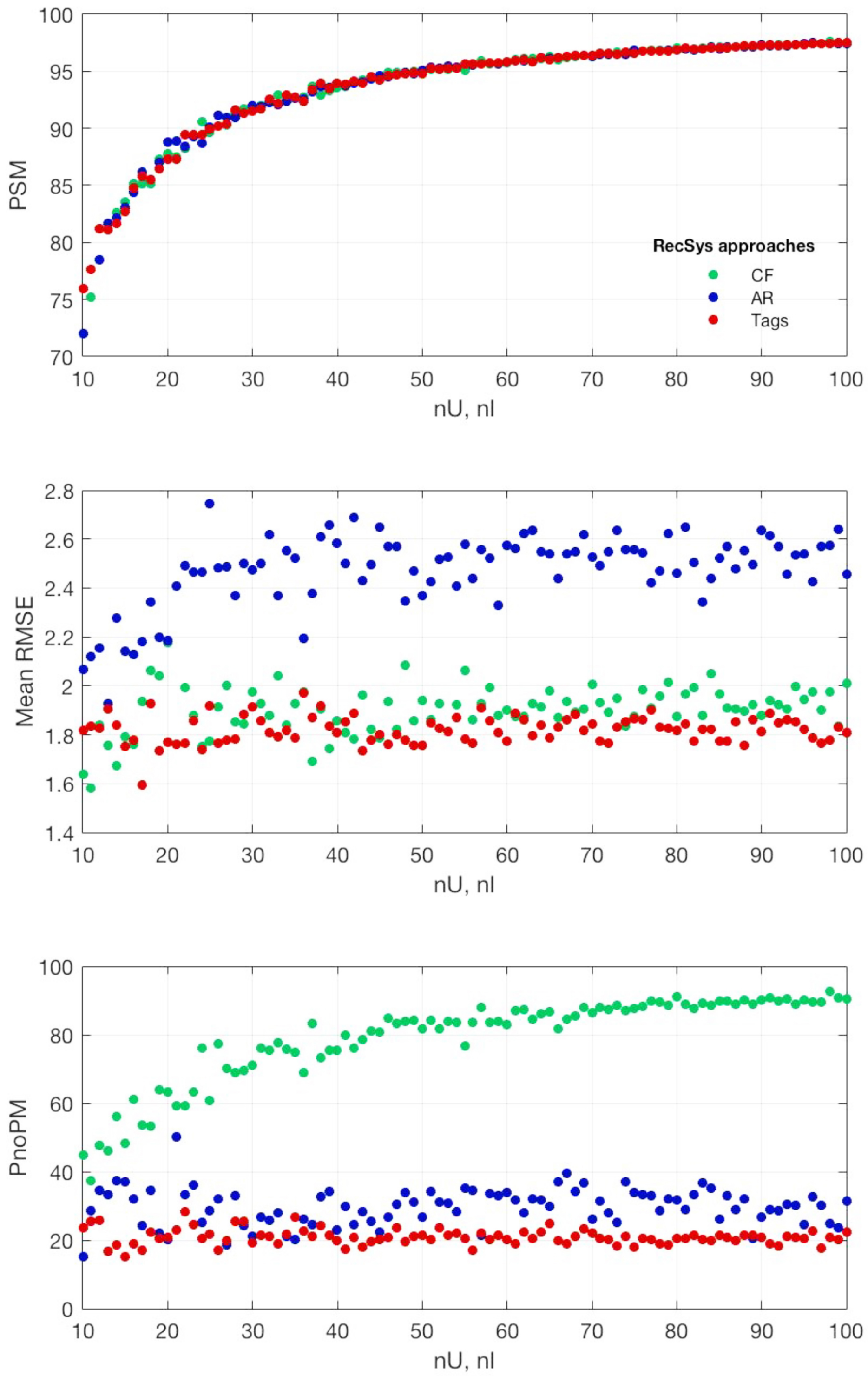

Figure 3: Dependence PSM, RMSE, PnoPM from nU, nI for NFB=[2 3], NTG=[9 12], fTG=[1 2]

Tag's approach with small number of feedbacks has:

- a quality of prediction better;
- a quantity of no-predicted values less.

Even more important in the fact that the time changes our actions. From this perspective we should care about the timeline of previous thoughts, how they are far from now, to exclude them and stay on the top of user's decisions right now.

References

[1] Barry N. Schwartz, *The paradox of choice: Why more is less*, HarperCollins Publishers, New York, 2004.

[2] Samia Beldjoudi, Hassina Seridi, C.F. Zucker, *Improving tag-based resource recommendation with association rules on folksonomies*, Second International Conference on Semantic Personalized Information Management: Retrieval and Recommendation, ACM, Aachen, Germany (2011), 26–37.

[3] Ido Guy, Naama Zwerdling, Inbal Ronen, David Carmel, Erel Uziel, *Social media recommendation based on people and tags*, 33rd international ACM SIGIR conference on Research and development in information retrieval, ACM, New York, NY, USA (2010), 194–201.

[4] Marius Kaminskas, Francesco Ricci, *Location-Adapted Music Recommendation Using Tags*, User Modeling, Adaption and Personalization, Springer Berlin Heidelberg (2011) **6787**, 183–240.

[5] Lyndon Kennedy, Malcolm Slaney, Kilian Weinberger, *Reliable tags using image similarity*, 1st workshop on Web-scale multimedia corpus, ACM, New York, NY, USA (2009), 17–24.

[6] Benjamin Markines, Ciro Cattuto, Filippo Menczer, Dominik Benz, Andreas Hotho, Gerd Stumme, *Evaluating similarity measures for emergent semantics of social tagging*, 18th international conference on World wide web, ACM, New York, NY, USA (2009), 641–650.

[7] Shilad Sen, Jesse Vig, John Riedl, *Tagommenders: Connecting Users to Items through Tags*, 18th international conference on World wide web, ACM, New York, NY, USA (2009), 671–680.

[8] Shiwan Zhao, Nan Du, Andreas Nauerz, Xiatian Zhang, Quan Yuan, Rongyao Fu, *Improved recommendation based on collaborative tagging behaviors*, 13th international conference on Intelligent user interfaces, ACM, New York, NY, USA (2008), 413–416.