Analysis, Design and Construction of a Recommendation System based on Principles of Neural Networks

Sofía R. Luque y Guzmán Sáenz
e-mail: solugusa@hotmail.com

Director: Juan Carlos García Díaz
e-mail: jcgarcia24@yahoo.com

Facultad de ingeniería de Sistemas
Escuela de Ciencias Naturales e Ingenierías
Telecomunicaciones y Tecnologías Web

Abstract

The recommendation systems are another alternative to the social process of recommendation. The way those systems work consists on asking the user to rank a group of items (films) which are presented to him. Those ranks are therefore cross-networked with other users which have ranked the films in a similar way to finally show the user series of recommend products. The Collaborative Filtering is a recommendation system which can be used both in users and in items. On this article we will use the CF (Collaborative Filtering) based on users, by means of using the Lineal Associative Memory (LAM). LAM is architecture founded on neuropsychological principles and is studied in the neural network community. The idea is to take randomly a group of users and with the application of CF estimate the items a user would choose according its preferences.

Key Words: Neural Network (NN), Recommendation Systems (RS), Collaborative Filtering by Lineal Memory (CFLM), Collaborative Filtering (CF).

1. Introduction

One of the most common techniques used today to recommend users is the Collaborative Filtering [2] (CF). The main objective of this technique is to suggest items to a particular user based on preferences and interests of other users with similar preferences and purchasing behaviour. The idea is to get the attention of this particular user on items that are not of their knowledge.

In order to get best results for the CF on some specific situations the Linear Associative Memory (LAM) has been implemented, founded on neuropsychological principles and which is expected to be more accurate when recommending an item. LAM is a type of neural network which is defined as the associative memory because records the initial information given by the user. This initial information can be retrieved providing similar data from different users who share similar preferences. The system is denominated lineal for the way its

network operates to make such recommendations of user's preferences. The alternative methodology LAM applied to the CF is known as CLAM, a system that explains how human beings make recommendations.

2. Content

2.1 Neural Networks

The neural networks [1] are nothing more than a way to simulate specific human characteristics such as the capacity to memorise and relate past actions [10]. If we try to analyse those problems that cannot be expressed through algorithms, we could realise that all of them have something in common: the expertise. A human being is able to find solutions based on cumulative expertise [9]. Therefore is clear that the best way to get closer to the problem will be to build systems which can be based on human behaviour.

2.2 Recommendation Systems

The ways these systems work consist basically on presenting a user series of questions which he has to weight according to his preferences. Therefore the system cross-networks his answers with other users who have similar preferences and finally the system present a recommended range of products. The recommendation system is built in two different phases [8]: Phase 1: The system asks the user its preferences. Phase 2: The system offers some recommendations to the users.

2.3 Collaborative Filtering

The CF is one of the most common techniques used today to make recommendations. The CF works collecting human preferences (known as ratings) for items in a given network and matching users who share same information needs and same preferences. Therefore, those users' preferences can help other users to decide what to consume [6]

The built of the CF technique is possible by following three steps [7]; first step: the built of user's profiles (rating), second step: the collaborative filtering locates people with similar profiles (neighbours) [3.5] and third step: the neighbours ratings are combined to make recommendations.

2.4 Associative Memory

The associative artificial memory or AM emulates how the brain works when the complete information of an item (i.e. image) can be reconstructed in the brain by fragments, parts or through a distortional version of the item.

Associative memories can be self-associative (AAM) or hetero-associative (HAM). On the first case the idea is to rebuild an entity represented by an item, based on the altered or incomplete version of that item, for that extent the training items are used to enter or the exit the network. On this case, naturally, the dimension of the exit and entry items is the same. In the case of the hetero-associative networks the idea is to obtain an exit item associated to an entry one, without being them together from the same dimensionality [4].

2.5 Linear Associative Memory

The easiest example of an associative memory is the linear associative memory LAM [4]. Those can be modelled as a kind of Feed Forward networks that have only one exit and therefore look in a lineal way for an exit vector which could be compatible to his exit vector. Figure No.1 shows the typical structure of this type of network. LAM memories are very simple to implement using lineal algebraically methods. It is possible to model the weight of the network as a weight matrix W M x *N where* (W) = $\{w_{ij}\}$, given w_{ij} the weight of the connection that comes from the entry unit i to the exit unit j, from a entry M network and N exits network.

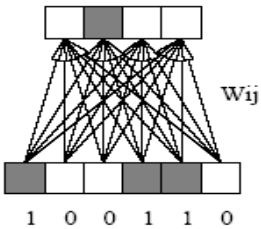


Figure 1. LAM structure

2.6 Collaborative filtering by linear associative memory (CLAM)

Since associative memory is modelled after biological memories, applying it to collaborative filtering forms a plausible first attempt at explaining how human minds make recommendations. In terms of implementation, we define the column vectors x^i *as patterns* to the system in which the *jth* element is:

$$x_j^i = \begin{cases} r_{i,j} - \overline{r}_i & r_{i,j} \neq \perp \\ 0 & r_{i,j} = \perp \end{cases}$$

And $r_{i,j}$ and r_i are user ratings as defined in section. The definition for x^a is analogous.

$$W = \sum_{i=1}^{m} \frac{x^i \cdot (x^i)^T}{\|x^i\|^\alpha}$$

The process of making recomendations is the same as that of recall. We calculate the column vector p^a as:

$$p^a = W x^a$$

We then recommend the top N items based on the elements of p^a, not including those items for which the active user has already rated.

3. Results

3.1 Methodology

The CLAM algorithms run some experiments on the MovieLens dataset. The dataset consists of 100.000 ratings, in the range of 1-5, from 943 users on 1682 movies. Each user has rated at least 20 movies, and each movie has been rated at least once. For each run of an experiment, we randomly divide the dataset such that 90% of the users are the training set and the other 10% are testing set. Each user in the testing set takes turn being considered the active user. For the active user, five of her ratings are randomly chosen to cero (0).

After having applied the collaborative algorithm to both the training set and the active user we finalise the process applying algorithm CLAM and then we have recommendation vector P which contains the Top 10 of the products recommended (the ten highest ranked films). This set is denoted I_N, but on this set we would not find films that the active user has no ranked yet.

For our evaluation metric, a maximum sum, **maxSum,** is calculated as the sum of all the positive withheld ratings for the active user.

$$maxSum = \sum_{j \in I_a^{withheld}} \max(0, x_{a,j})$$

Another value, **SumNoPenalty,** is the sum limited to the positive withheld ratings from products returned by the Top N, that is:

$$sumNoPenalty = \sum_{j \in \{I_a^{withheld} \cap I_N\}} \max(0, x_{a,j})$$

Having the results of these two equations we then need to find the NoPenalty value, which shows a 100% of all the films which have been recommended on top the N with positive scores. On the other hand we also calculate a penalised sum, sumPenalty, which is analogous to sumNoPenalty but includes the negative ratings of products in the top N:

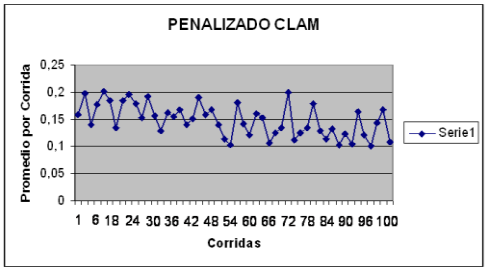
$$sumPenalty = \sum_{j \in \{I_a^{withheld} \cap I_N\}} x_{a,j}$$

Since our ranking task also takes into account the popularity of movies, we need to ensure that popularity is not an overwhelming factor and that personalisation still plays a significant role. To investigate this bias, we examined a naive algorithm that recommends the top N movies (that the active user has not rated) based on just the sum of all trainings user's ratings.

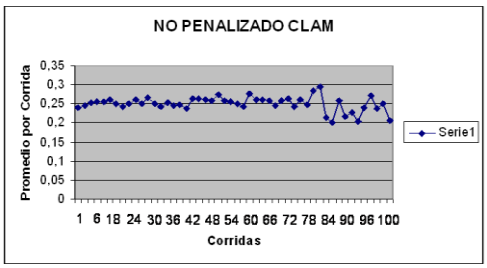
"NAIVE" is a simple method of recommendation which is impersonalised and just attempts to recommend products more active users have given higher ratings to.

$$p_j^a = \sum_{i: r_{i,j} \neq \perp} (r_{i,j} - \overline{r}_i)$$

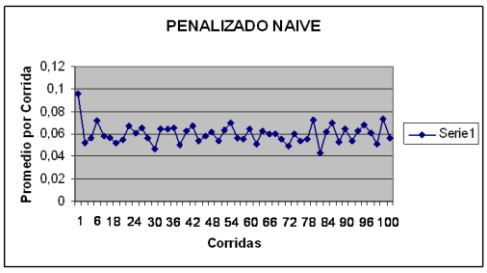
To demonstrate the enhancement of each of the algorithms, there were 100 runs with the same active users for both the algorithm CLAM and for the "NAIVE" method, having the following results:



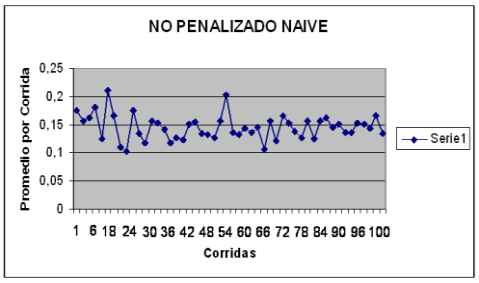
Graph 1. Penalty Values CLAM



Graph 2. No Penalty values CLAM



Graph 3. Penalty values NAIVE



Graph 4. No Penalty values NAIVE

Each experiment is the result of 100 runs. Finding the average and standard deviation in brackets. We have taken α =2 since normalising both input and output is standard practice in associative memory and it performs better than other settings compared with other values α [4]

	PROMEDIOS – DESVIACIÓN ESTANDAR	
	Penalty	**NoPenalty**
NAIVE	0.060037(0.008)	0.145714(0.00219)
CLAM (α = 2)	0.148374(0.029)	0.250468(0.01860)

As we can see on the Chart, the algorithm CLAM have a significant better performance than that of the No personalised "NAIVE" taking into account both metrics: Penalty and NoPenalty. However, the improvement on the NoPenalty metric is more pronounced because its standard deviation is smaller.

Conclusions

Conclusions

It was implemented the filtering collaborative method based on lineal memory self-associative, which is the first step on the explanation on how the human brain execute the recommendation function. We compared the CLAM algorithm with the NAÏVE method, demonstrating that CLAM algorithm have a significant better performance when compared with the non-personalized method NAÏVE for the dataset MovieLens.

The CLAM method is built over a symmetric matrix of n x n (n is the number of items). Even though it is independently from the number of users, is slightly demanding memory and processing when n is big.

It is recommended to compare against other traditional methods such as those analyzed on the article "A comparison of Several Predictive Algorithms for Collaborative Filtering on Multi-Valued Ratings", and then to try to locate the method in an objective way, inside the growing group of algorithms used for recommendation systems. Finally, it is recommended to use standard techniques in order to reduce the dimensionality, such as the Analysis of Principal Components and the magnitude pruning.

Autores

Sofía R. Luque y Guzmán Sáenz, estudiante de la Facultad de Ingeniería de Sistemas de la Universidad Autónoma de Bucaramanga (UNAB). Colombia-Santander.

5. Referencias

[1] A. Alegre López. "Simulador de Redes Neuronales". Tesis de Pregrado.
Argentina, 2003.

[2] C. González Caro; M. Calderón Benavides; J. Pérez Alcázar; J. García Díaz y J. Delgado. "A comparison of Several Predictive Algorithms for Collaborative Filtering on Multi-Valued Ratings". Symposium on Applied Computing, Proceedings of the 2004 ACM symposium on Applied computing. Pa. 1033-1039. ISBN 1-58113-812-1, 2004.

[3] C. González Caro; M. Calderón Benavides; J. Pérez Alcázar; J. García Díaz y J. Delgado. Towards a More Comprehensive Comparison of Collaborative Filtering Algorithms. Proceedings of the 9th International Symposium on String Processing and Information Retrieval, p.248-253, September 11-13, 2002 **.**

[4] Ch. Lam. Collaborative Filtering using Associative Neural Memory, Stanford University, USA. Publicado en Semantic Web Personalization, AAAI Workshop on, The Nineteenth National Conference on Artificial Intelligence - AAAI 2004 , July 25-26, San Jose, California, 2004.

[5] Filtrado Colaborativo. http://maxft.udc.ie/chap4.html

[6] J. Breese; D. Heckerman y C. Kadie. Empirical Analysis of Predictive Algorithms for Collaborative Filtering. Microsoft Research, Redmond WA, 98052-6399, USA.1998.

[7] J. Herlocker; J. Konstan; A. Bochers and J. Riedl. An algorithmic framework for performing collaborative filtering. In SIGIR'99: Procceding of the 22nd Annual International ACM SIGIR Conference on Research and Development in information Retrieval. 1999.

[8] J. Herlocker; J. Konstan; A. Bochers and J. Riedl. Explaining Collaborative Filtering Recommendations. http://cs.oregonstate.edu/~herlock/papers/explanations_cscw2000.pdf. 2000.

[9] Hassoun Fundamentals of Artificial Neural Networks. The MIT Press. 1995.

[10] Pennock, Horvitz Social choice theory and recommender systems: Analysis of the... et al. – 2000

[11] Redes Neuronales artificiales - Artificial Neural Networks http://www.gc.ssr.upm.es/inves/neural/ann2/anntutorial.html

[12] Redes de neuronas artificiales y pensamiento http://personal5.iddeo.es/wintrmute/ia/neuronal.htm