Proceedings of the 5th WSEAS Int. Conf. on Instrumentation, Measurement, Circuits and Systems, Hangzhou, China, April 16-18, 2006 (pp285-290)

A Recommender System Framework combining Neural Networks & Collaborative Filtering

CHARALAMPOS VASSILIOU DIMITRIS STAMOULIS DRAKOULIS MARTAKOS
Department of Informatics and Telecommunications
National and Kapodistrian University of Athens
Panepistimioupolis, 157-84, Athens
GREECE

SOTIRIS ATHANASSOPOULOS
Department of Physics, Nuclear & Particle Physics Division
National and Kapodistrian University of Athens
Ilissia, 157-71, Athens
GREECE

Abstract: - Most recommender systems use collaborative filtering or content-based methods to predict new items of interest for a user. While both methods have their own advantages, individually they fail to provide good recommendations in many situations. An alternative method to content-based filtering could be the use of neural networks which also incorporate the essence of progressive learning as this filtering method is increasingly used by a system. Incorporating components from both methods, a hybrid recommender system can overcome these shortcomings. In this paper, we present an elegant and effective framework for combining neural networks and collaborative filtering. Our approach uses a neural network to recognize implicit patterns between user profiles and items of interest which are then further enhanced by collaborative filtering to personalized suggestions. Our preliminary study indicates that this hybrid approach is particularly promising when compared to pure content-based or collaborative filtering methods.

Key-Words: - Hybrid Personalization, Recommendation Engine, Neural Networks, Collaborative Filtering

1 Introduction

As soon as the research society as well as the commercial users started understanding the potential of web technologies for one-to-one marketing, otherwise named the mass customization capacity, recommender systems to personalize content appeared. Obviously enough, two main venues of thought emerged. One, collaborative filtering methods [11], is based on the hypothesis that similar users will demonstrate similar online behavior, and, therefore, what one is interested in will most probably be of interest to a similar user. The similarity of users is based upon user profiles. The other category of methods, content based [ibid.] takes into account the similarity of items, rather than users, in order to propose to the user 'a closest match'.

Each set of methods has its own advantages and disadvantages, or to put it differently, provides better results under different circumstances. That's why combinations of methods have started appearing in the relevant literature. Such an analysis [12] has investigated the issue of whether any of these methods are more appropriate for a particular phase of the customer decision process, if an online interaction is a sale. This paper had shown that content based personalization methods had better be used during the pre-purchase phase of the customer decision process, observational based methods [9] for the purchase phase and collaborative filtering methods for the post-purchase phase. Looking further into the possible alternatives of combining personalization methods, the two main types of collaborative techniques, memory-based and model-based algorithms, have been combined into a hybrid architecture [13]. Experimental results of the hybrid architecture have successfully verified its increased personalization effectiveness over single collaborative filtering techniques.

Studying the various personalization methods, it became evident that a learning perspective is missing from them. To this end, neural networks were thought of a necessary component to be added into a personalization architecture, because of their learning capabilities. During the last decade, artificial neural networks have been utilized to construct predictive statistical models in a variety of scientific problems ranging from astronomy to experimental high-energy physics to protein structure [3], [4]. In a typical application, a multilayer feed-forward neural network is trained with back-propagation or some other supervised training algorithms [6], [5], [7], [2] so as to create a "predictive" statistical model of a certain input–output mapping, which may in general be physical or mathematical in character. Information contained in a set of learning examples of the input–output association is embedded in the weights of the connections between the layered units. This information may (or may not) be sufficient to allow the trained network to make reliable predictions for examples outside the learning set. At any rate, the network is taught to generalize (well or poorly), based on what it has learned from the set of examples. In the more mundane language of function approximation, the neural-network model provides a means for interpolation or extrapolation.

There are a few different ways of interconnecting neural networks to a personalization method or a set of them. In this paper, a neural network is used to recognize implicit patterns between user profiles and items of interest, which are then further enhanced by collaborative filtering to personalized suggestions. Our preliminary study indicates that this hybrid approach is particularly promising when compared to pure content-based or collaborative filtering methods. Section two provides a technical description of neural networks, of the collaborative filtering techniques and one of the possible interconnections between neural networks and the collaborative filtering algorithms. Section three discusses a few real problems met during the course of this on-going research and finally, the conclusions provide some food for further thought.

2 Framework Description

Our immediate tasks are to provide a (i) system overview by identifying its building blocks and specify (ii) the structure, unit-dynamics and training algorithm of the neural network that will be developed to model the existing data. We must also specify the (iii) collaborative filtering algorithm that will decode the patterns comprehended by the neural network to personalized predictions and (iv) the technique used for the interconnection and details of our hybrid approach.

2.1 System Overview

The general overview of our system and the connection between the Neural Network and the Collaborative Filtering algorithms is shown in the following figure and follows closely the approach presented in [10].

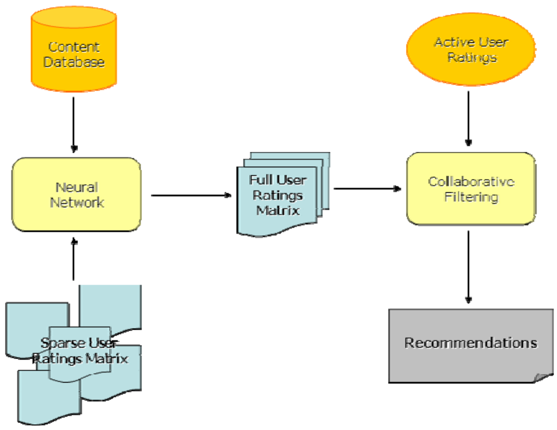

Fig. 1: System Overview

Any given web site offering this hybrid personalization approach stores all related content (i.e. multimedia objects, products, documents, etc.) within its content database. The corresponding dataset of this web site also provides the user-ratings matrix, which is a matrix of users versus items, where each cell is the rating given by a user to an item. We will refer to each row of this matrix as a user-ratings vector.

The user-ratings matrix is very sparse, since most items have not been rated by most users. The neural network algorithm is trained on each user-ratings vector and a pseudo user-ratings vector is created.

A pseudo user-ratings vector contains the user's actual ratings and neural network predictions for the unrated items. All pseudo user-ratings vectors put together form the pseudo ratings matrix, which is a full matrix. Now given an active user's[1] ratings, predictions are made for a new item using CF on the full pseudo ratings matrix.

2.2 Neural Network

[1] The active user is the user for whom predictions are being made.

A multilayer feed-forward architecture is adopted, with various numbers of hidden layers and distributions of units among layers. The connection from unit m to unit n is characterized by a real-number weight w_{mn} with initial value positioned at random in the range [−1, 1]. When a pattern μ is impressed on the input interface, the activities of the input units propagate through the entire network. Each unit in a hidden layer or in the output layer receives a stimulus $u_n = \sum_m W_{mn}\alpha_m$, where the a_m are the activities of the units in the immediately preceding layer. The activity of generic unit m in the hidden or output layers is in general a nonlinear function of its stimulus, $\alpha_m = g(u_m)$. In our work, the unit activation functions $g(u)$ are selected between the logistic (sigmoid), hyperbolic tangent and linear forms. The system response may be decoded from the activities of the units of the output layer while the dynamics is particularly simple: the states of all units within a given layer are updated successively, proceeding from input to output.

Several training algorithms exist that seek to minimize the cost function with respect to the network weights. For the cost function we make the traditional choice of the sum of squared errors calculated over the learning set, or more specifically

$$E = \sum_{\mu} E^{(\mu)} = \frac{1}{2}\sum_{\mu,i}(t_i^{(\mu)} - o_i^{(\mu)})^2 \tag{1}$$

where $t_i^{(\mu)}$ and $o_i^{(\mu)}$ denote, respectively, the target and actual activities of unit i of the output layer for input pattern (or example) μ. The most familiar training algorithm is standard back-propagation [6], [5] (hereafter often denoted SB), according to which the weight update rule to be implemented upon presentation of pattern μ is

$$\Delta w_{mn}^{(\mu)} = -\eta\frac{\partial E^{(\mu)}}{\partial w_{mn}} + \alpha\Delta w_{mn}^{(\mu-1)} \tag{2}$$

where η is the learning rate, α is the momentum parameter, and $\mu-1$ is the pattern impressed on the input interface one training step earlier. The second term on the right-hand side, called the momentum term, serves to damp out the wild oscillations in weight space that might otherwise occur during the gradient-descent minimization process that underlies the back-propagation algorithm. Our artificial neural networks are trained with a modified version of the SB algorithm [1] that we have found empirically to be advantageous in the majority of problems. In this algorithm, denoted MB, the weight update prescription corresponding to Eq. (2) reads

$$\Delta w_{mn}^{(\mu)} = -\eta\frac{\partial E^{(\mu)}}{\partial w_{mn}} + \alpha S_{mn}^{(\mu-1)} \tag{3}$$

the momentum term being modified through the quantity

$$S_{mn}^{(\mu-1)} = \frac{S_{mn}^{(\mu-2)}e + \Delta w_{mn}^{(\mu-1)}}{e+1} \tag{4}$$

In the latter expression, e is the number of the current epoch, with $e = 0,1,2,3,...$The replacement of $\Delta w_{mn}^{(\mu-1)}$ by $S_{mn}^{(\mu-1)}$ in the update rule for the generic weight w_{mn} allows earlier patterns of the current epoch to have more influence on the training than is the case for standard back-propagation. By the time e becomes large, $S_{mn}^{(\mu-1)}$ is effectively zero. It can be shown, after rather lengthy algebra, that if a plateau region of the cost surface has been reached (i.e., $\frac{\partial E}{\partial w_{mn}}$ remains almost constant) and e is relatively large, then Eq. (3) converges to

$$\Delta w_{mn}^{(\mu)} = -2\frac{1}{1-\alpha}\frac{\partial E^{(\mu)}}{\partial w_{mn}} \tag{5}$$

thus achieving an effective learning rate twice that of the SB algorithm (cf. [6]).

Further information on the MB algorithm together with various techniques regarding the coding of input and output interfaces and the training methodology can be found in [1].

2.3 Collaborative Filtering

We implemented a pure collaborative filtering component that uses a neighborhood-based algorithm [8]. In neighborhood-based algorithms, a subset of users are chosen based on their similarity to the active user, and a weighted combination of their ratings is used to produce predictions for the active user. The algorithm we use can be summarized in the following steps:

1. Weight all users with respect to similarity with the active user. Similarity between users is

Proceedings of the 5th WSEAS Int. Conf. on Instrumentation, Measurement, Circuits and Systems, Hangzhou, China, April 16-18, 2006 (pp285-290)

measured as the Pearson correlation between their ratings vectors.

2. Select n users that have the highest similarity with the active user. These users form the neighborhood.
3. Compute a prediction from a weighted combination of the selected neighbors' ratings. In step 1, similarity between two users is computed using the Pearson correlation coefficient, defined below:

$$P_{a,u} = \frac{\sum_{i=1}^{m}(r_{a,i} - \bar{r}_a) \times (r_{u,i} - \bar{r}_u)}{\sqrt{\sum_{i=1}^{m}(r_{a,i} - \bar{r}_a)^2 \times \sum_{i=1}^{m}(r_{u,i} - \bar{r}_u)^2}} \qquad (6)$$

where $r_{a,i}$ is the rating given to item *i* by user *a*; r_a is the mean rating given by user *a*; and *m* is the total number of items.

In step 3, predictions are computed as the weighted average of deviations from the neighbor's mean:

$$p_{a,i} = \bar{r}_a + \frac{\sum_{u=1}^{n}(r_{u,i} - \bar{r}_u) \times P_{a,u}}{\sum_{u=1}^{n} P_{a,u}} \qquad (7)$$

where $p_{a,i}$ is the prediction for the active user *a* for item *i*; $P_{a,u}$ is the similarity between users *a* and *u*; and *n* is the number of users in the neighborhood. For our experiments we used a neighborhood size of 30, based on the recommendation of [8], [10]. It is common for the active user to have highly correlated neighbors that are based on very few co-rated (overlapping) items. These neighbors based on a small number of overlapping items tend to be bad predictors. To devalue the correlations based on few co-rated items, we multiply the correlation by a *Significance Weighting* factor [8]. If two users have less than 50 co-rated items we multiply their correlation by a factor $sg_{a,u} = n/50$, where *n* is the number of co-rated items. If the number of overlapping items is greater than 50 then we leave the correlation unchanged i.e. $sg_{a,u} = 1$.

2.4 Connecting Neural Network and Collaborative Filtering Algorithms

In the proposed hybrid combination of neural network and collaborative filtering algorithms, we first create a pseudo user-ratings vector for every user *u* in the database. The pseudo user-ratings vector, v_u, consists of the item ratings provided by the user *u*, where available, and those predicted by the neural network predictor algorithm otherwise.

$$v_{u,i} = \begin{cases} r_{u,i} : & \text{if user } u \text{ rated item } i \\ c_{u,i} : & \end{cases}$$

In the above equation $r_{u,i}$ denotes the actual rating provided by user *u* for item *i*, while $c_{u,i}$ is the rating predicted by the neural network system.

The pseudo user-ratings vectors of all users put together give the dense pseudo ratings matrix *V*. We now perform collaborative filtering using this dense matrix. The similarity between the active user *a* and another user *u* is computed using the Pearson correlation coefficient described in Eq. 6. Instead of the original user votes, we substitute the votes provided by the pseudo user-ratings vectors v_a and v_u.

3 Current Study and Evaluation Methodology

At the current stage of our research we are conducting a set of evaluation experiments based on large datasets aimed specifically for applying personalization techniques and providing recommendations. Originally we have used the MovieLens[2] dataset provided by the GroupLens Research Project[3]. The GroupLens Research Project is a research group in the Department of Computer Science and Engineering at the University of Minnesota. Members of the GroupLens Research Project are involved in many research projects related to the fields of information filtering, collaborative filtering, and recommender systems.

The MovieLens dataset that we originally used consists of 100,000 ratings for 1682 movies by 943 users. The data we use from the MovieLens dataset were: user (age, gender, occupation, zipcode), movie title, genre and user ratings per items. Despite the large number of user ratings, during the initial stage of the neural network processing the data proved to be incoherent for the neural network predictor algorithm to learn based on the user profile. After a preliminary OLAP analysis of the MovieLens dataset proved that all data-views of the

[2] http://movielens.umn.edu/login
[3] http://www.grouplens.org/

user and movie data provided the same results, thus making it impossible for the neural networks predictor algorithm to learn patterns and extract predictions. Similar problems appeared within the CF processing of the data prior to any neural network processing.

Despite the insufficiency of the MovieLens dataset, we continue to work in the domain of movie recommendations in order to demonstrate the performance of our hybrid approach by using the EachMovie[4] dataset. HP/Compaq Research (formerly DEC Research) ran the EachMovie movie recommender. When EachMovie was shutdown, the dataset was available to the public for use in research. MovieLens was originally based on this dataset. It contains 2,811,983 ratings entered by 72,916 for 1628 different movies, and it has been used in numerous CF publications. As of October, 2004, HP retired the EachMovie dataset. The EachMovie dataset contains rating data provided by each user for various movies. User ratings range from zero to five stars. Zero stars indicate extreme dislike for a movie and five stars indicate high praise. To have a quicker turn-around time for our experiments, we will only use a subset of the EachMovie dataset. This dataset contains 7,893 randomly selected users and 1,461 movies for which content is available from the Internet Movie Database (IMDb)[5].

The experimental methodology to be followed the guidelines of [10] and aims to compare the hybrid approach proposed in this paper to a pure neural networks algorithm, a collaborative filtering algorithm, and a naive hybrid approach. The naive hybrid approach will take the average of the ratings generated by the pure neural networks algorithm and the pure collaborative filtering algorithm. For the purposes of comparison, we will use a subset of the ratings data from the EachMovie data set (described above). Fifteen percent of the users will be randomly selected to be the test users. From each user in the test set, ratings for 30% of items will be withheld. Predictions will be computed for the withheld items using each of the different algorithms. The quality of the various prediction algorithms will be measured by comparing the predicted values for the withheld ratings to the actual ratings.

[4] http://research.compaq.com/SRC/eachmovie
[5] http://www.imdb.com

4 Conclusion

This paper aimed to bring forward the need to combine collaborative filtering techniques for personalization with neural networks, that possess the ability to learn / adapt. Previous research efforts have shown that the effectiveness of personalization methods can be increased by their combination. Although no solid data from real-world experiments can be given for the time being, there are positive preliminary indications that interconnecting collaborative filtering with neural networks increases the effectiveness of the personalization process. This is because of the learning capability of the neural networks, which can be used to leverage the capabilities of collaborative filtering techniques for successful recommendations.

Next steps into this research have to do with fuzzy logic, since recommendations are inherently not of crisp nature. Meaning that the user to whom the recommendations are addressed may express his/her view on the usefulness of the recommendations with relative measures of acceptance / satisfaction, such as ‘pretty good’, ‘interesting’, ‘not so interesting’ etc. By assigning fuzzy values to recommendations in association with observational personalization methods, new research areas will be opened in exploring the issue of successful recommender systems.

Acknowledgments:
The European Commission and the Greek General Secretariat of Research and Technology have funded this work through a specific category of projects that focuses on supporting new researchers and new advanced IT developments. The authors wish to acknowledge these bodies for their support.

References:

[1] S. Athanassopoulos, E. Mavrommatis, K.A. Gernoth and J.W. Clark. Nucl. Phys. A743 (2004) 222 – 235.
[2] C. Bishop, Neural Networks for Pattern Recognition, Clarendon, Oxford, 1995.
[3] V. Cherkassky, J.H. Friedman, W.Wechsler (Eds.), From Statistics to Neural Networks. Theory and Pattern Recognition Applications, Springer-Verlag, Berlin, 1994.
[4] W. Clark, T. Lindenau, M.L. Ristig (Eds.), Scientific Applications of Neural Nets, Springer-Verlag, Berlin, 1999.
[5] J.W. Clark, Phys. Med. Biol. 36 (1992) 1259.

[6] J. Hertz, A. Krogh, R.G. Palmer, Introduction to the Theory of Neural Computation, Addison–Wesley, Redwood City, CA, 1991.

[7] S. Haykin, Neural Networks: A Comprehensive Foundation, McMillan, New York, 1998.

[8] Herlocker, J.; Konstan, J.; Borchers, A.; and Riedl, J. 1999. An algorithmic framework for performing collaborative filtering. In SIGIR '99: Proceedings of the 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, 230–237.

[9] Mulvenna, M.D., Ananad, S.S. & Buchner, A.G. 2000 Personalization on the Net using Web Mining. Communications of the ACM, 43(8), 122-125.

[10] Melville, P., Mooney, R. J., and Nagarajan, R. 2002. Content-boosted collaborative filtering for improved recommendations. In Eighteenth National Conference on Artificial intelligence (Edmonton, Alberta, Canada, July 28 - August 01, 2002). R. Dechter, M. Kearns, and R. Sutton, Eds. American Association for Artificial Intelligence, Menlo Park, CA, 187-192.

[11] Ridell, J. et. Al. 2002. Word of mouse: the marketing power of collaborative filtering. New York. Warner Books.

[12] Vassiliou, Ch., Stamoulis, D.S., Spiliotopoulos, A., Martakos, D. 2003 Creating Adaptive Web Sites using Personalization techniques. In Adaptive Evolutionary Information Systems, editor N. V. Patel, Idea Publishing Group, p. 261 – 285.

[13] Vassiliou, Ch., Stamoulis, D.S., Martakos, D. 2006 A hybrid content-based clustering architecture: minimising uncertainty in personalised multimedia content. To appear in International Journal of Intelligent Systems Technologies and Applications (IJISTA).