Weightless Neural Network WiSARD Applied to Online Recommender Systems

Conference Paper · October 2018

DOI: 10.1109/BRACIS.2018.00067

3 authors:


Kleyton Pontes Cotta
Federal University of Rio de Janeiro


Raul Sena Ferreira
Federal University of Rio de Janeiro


Felipe M.G. França
Federal University of Rio de Janeiro

Weightless Neural Network WiSARD Applied to Online Recommender Systems

Kleyton Pontes Cotta
COPPE - Graduate School and Research in Engineering
Universidade Federal do Rio de Janeiro (UFRJ)
Rio de Janeiro, Brazil
Email: kpcotta@cos.ufrj.br

Raul S. Ferreira
COPPE - Graduate School and Research in Engineering
Universidade Federal do Rio de Janeiro (UFRJ)
Rio de Janeiro, Brazil
Email: raulsf@cos.ufrj.br

Felipe M.G. França
COPPE - Graduate School and Research in Engineering
Universidade Federal do Rio de Janeiro (UFRJ)
Rio de Janeiro, Brazil
Email: felipe@cos.ufrj.br

***Abstract*—Recommender systems generally are made to predict user preferences' for items. However, in high dimensional datasets this task demands high computational costs. Taking into account that data distribution changes through time, it is important that online recommender systems have a fast retraining process in order to keep the model updated, delivering accurate predictions. Therefore, we propose a new approach for recommender systems using a weightless neural network, denominated WiSARD. We show that our proposal increases training and prediction processing speed, without decreasing the quality of predictions. First results show that our proposal is 306% faster than the improved regularized singular value decomposition (IRSVD), a well-known state-of-the-art algorithm. Moreover, our proposal still had an improvement of 3.7% regarding the mean absolute error (MAE). We show how to apply the WiSARD algorithm for online recommender systems, its drawbacks, and insights for further research.**

Keywords—Online Recommender Systems; Weightless Neural Network; WiSARD.

I. Introduction

Recommender systems were primarily created to suggest personalized items based on user interests'. For example, it can minimize the possibility of occur an abandonment during a buying process since making choices with several options are uncomfortable and difficult [1]. Users generally recommend products and these information about the recommendations are stored in the system. Hence, these historical data becomes the base to form groups of potential candidates for consuming an item [2]. However, it is common for these applications to bring irrelevant data [3]. Thus, it is necessary to incorporate information filtering methods with the purpose to deliver interesting content. In general, there are three information filtering approaches: content-based filtering, collaborative filtering and hybrid filtering (mix of both).

Content-based filtering assumes that users tend to be interested for similar items to those who have shown interest in the past [4]. On the other hand, collaborative filtering does not require understanding or knowledge about the items. For its flexibility, is one of the most popular approaches and are very common on the internet [5]. Collaborative filtering aims to recommend an item to an user based on items previously evaluated by other users.

Real-time recommender systems needs to tackle challenges such as real-time updating and changes in the distribution through the time [6]. Since data streams comes in high velocity, a recommender system needs to update and response instantaneously in order to catch users' instant intention and demands. Besides, as stated by a recent survey [7], online recommender systems suffer from problems such as cold start [8], scalability [9] and sparsity [10]. Therefore, to overcome the mentioned problems, we propose the use of weightless neural networks [11].

Unlike traditional networks, the learning process of a weightless neural network (WNN) is accomplished by modifying the words stored in the memory. Thus, it allows the construction of flexible and fast learning. The inputs and the outputs neurons are binary numbers and there are no weights between the neurons. WNN has the ability to learn with only one presentation of training pattern, improving the performance over sparse data [12]. Moreover, these networks have simple structure, and a powerful classifier fast enough to handle non-linear problems [13]. Besides, WNN are subject of recent and intensive research in several domains such as, change detection in the field of visualization [14], multilingual part-of-speech tagging mechanism [15], background estimation for video scene applications [16], fault detection and diagnosis in univariate and multivariate dynamic systems [17], open set recognition [18] and so on.

A popular WNN is the Wilkes, Stonham and Aleksander Recognition Device (WiSARD) [19]. This neural network was designed for pattern recognition, originally applied for image applications, implemented in hardware giving real-time performance [19]. WiSARD is a network where neurons are implemented with random access memories (RAM). The WiSARD knowledge is directly stored in the node's memory during learning and can be modeled as n-tuple classifier [20]. Thus, due to its simplicity, WiSARD is capable to operate at high speed. Moreover, the method is easily parallelized, improving the scalability [12].

However, despite the extensive recent research, there is no work related to WiSARD applied to recommender systems. Therefore, in this work, we propose to use WiSARD as an alternative in online recommender systems, overcoming problems in collaborative-filtering algorithms such as low processing time, and low scalability, without loose quality in the predictions. Thus, this work is organized as follows: Section II presents the proposal for predicting ratings. Section III presents the basic concepts of recommendation system. Section IV shows the methodology whereas Section V detail the experiments and the results. Finally, Section VI summarizes the strengths and weaknesses of our proposal and insights for further research.

II. The WiSARD n-tuple Classifier

As previously mentioned, WiSARD is a network where neurons are implemented with random access memories (RAM). A RAM neural network is a 1-WNN in which the neurons are 1-RAMs. The I-RAM is just called a RAM node. The activation function is the identity function. There are 2^{2n} different functions applied to n address corresponding to 2^n RAM states. Thus, a single RAM computes all binary functions of its input while a weighted network only computes linearly separable functions of its input [20]. Despite RAM learns with only one presentation of training pattern at time, several RAM networks can be combined and each network trained according to a pattern class. These modified RAMs group are denominated discriminator [20].

WiSARD architecture is composed by a group of discriminators, each one corresponding to a class problem to be mapped. During the training phase, the input pattern is presented only for the discriminators with their corresponding target class. The classification is obtained by presenting the input pattern for all discriminators, selecting the class with the highest activation value. This approach provides an advantage when the network needs to be trained in a very short time. For training a WNN is necessary to increase the value of each RAM represented by the number of bits of the input. In a network without training phase, all discriminators are initialized with zero. Figure 1 illustrates this model, where r is the amount of RAM for each discriminator and d is the number of discriminators. A RAM learning algorithm performs the following five steps [20]:

1) Present an input pattern in the terminals;
2) Select the RAM nodes which should learn and present the desired output to the vector d;
3) Set the write enable in terminals that will learn, storing the desired outputs in the memory positions;
4) Goes back to the first step for all N training patterns;
5) The algorithm stops when the error tolerance is reached and return to the first step.

WISARD is based on RAM nodes with k ($2 <= k <= 16$) inputs. Each of the k inputs of a RAM node takes a binary value using a random but fixed mapping. Thus, a discriminator represents a class of objects to be recognized. A WISARD system consists of a number of discriminators and a decision unit which calculates the responses and the confidence of the discriminators during the recognition process. Besides, all the storage locations of discriminators are set to 0 before the training process. The classification selects a particular discriminator to be trained.

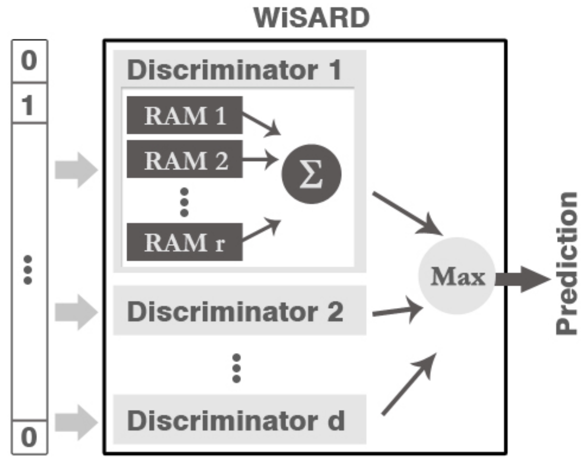


Fig. 1. WiSARD model.

III. Recommender systems

Recommender systems are techniques and tools used to suggest personalized items based on user interests. In a typical recommender system, the information captured by the users are the input of the system. The output are recommendations of the items that meet the expectations and needs of the users [2]. Thus, recommender systems are designed to filter information according to users' interests helping them to deal with irrelevant content [3].

One of the first groups to work with recommender system based on collaborative filtering was the Movie Lens group system [21]. They applied Pearson's correlation [22] to identify similarity between users. Among the several recommender systems developed during the years we can outline the improved regularized singular value decomposition (IRSVD) algorithm [23]. It uses an approach where the model extracts latent factors through matrix factorization algorithms [24]. The IRSVD algorithm is a weighting approach applied to the regularized SVD model. Equation 1 shows how the regularized SVD makes predictions for an user i and a movie j.

$$\hat{y}_{ij} = u_i^T v_j \tag{1}$$

Parameters u_i and v_j are K-dimensional vectors of parameters and are estimated minimizing the sum of squared residuals, one at a time, using gradient descent with regularization and early stopping methods. However, IRSVD add biases to the regularized SVD model: one parameter c_i for each user and one parameter d_j for each movie, as explained in Equation 2.

$$\hat{y}_{ij} = c_i + d_j + u_i^T v_j \tag{2}$$

Weights c_i, d_j are trained simultaneously with u_{ik} and v_{jk}. Besides, three parameters are set: lrate = 0.001, λ_2 = 0.05 and global_mean = 3.6033, as explained in Equation 3 and 4.

$$c_i += lrate * (r_{ij} - \lambda_2(c_i + d_j - global\_mean)) \tag{3}$$

$$d_j += lrate * (r_{ij} - \lambda_2(c_i + d_j - global\_mean)) \tag{4}$$

The performance of IRSVD is superior front of a traditional baselines such as nearest neighbor techniques [25], for producing product recommendations, allowing the incorporation of additional information such as implicit feedback, temporal effects, and confidence levels [24]. Moreover, this algorithm is popular for winning a competition promoted by Netflix [1].

IV. Methodology

We split the solution in three parts: feature selection, binarization and WiSARD processing. The first part is related to the chosen attributes and the processing made on top of these attributes. The second phase is responsible for transforming data into a representation that can be understood by the WiSARD model. Finally, the third phase is related to the necessary steps to WiSARD learn with data for posterior prediction. The methodology for the experiments were executed in the following order, illustrated in Figure 2.

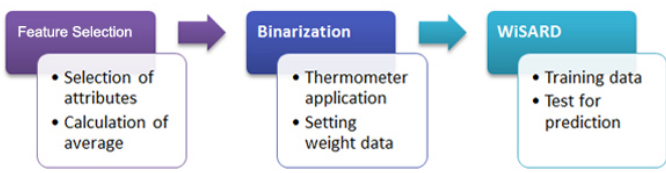


Fig. 2. Model Development.

For the feature selection were chosen seven attributes which the first five attributes are regarding movie and the two remaining attributes are related to the user:

- Genre;
- Produced year;
- Average rating;
- Average score given by the user;
- Average grade given a user and genre;
- Age;
- Gender.

These attributes were empirically chosen aiming to validate an initial set of parameters in the model. Worth to mention that other attributes could be tested but here the intention is to validate how the WiSARD network can be applied for recommender systems.

A. Binarization

For the binarization phase, we applied a thermometer method. The thermometer method is built to create a bit string using d bits, simulating a scale. The thermometer encoding considers the Hamming distance [26] between patterns [27]. Thus, we proportionally transform a quantitative variable into values composed by a sequence of zeros and ones. Despite thermometer increase the number of features its size has low negative impact in WiSARD regarding the speed [27].

[1]https://www.netflix.com/br/

Therefore, we applied a thermometer in the year of the movie, on the user's age and the averages. In neural networks, a thermometer has the objective to sequentially order the units according to the original data. This increases the value of a unit inside a sequence, as shown in Table I.

TABLE I. Thermometer codification for quantitative variables.

Age	Codification
0-9	0000000
10-19	1000000
20-29	1100000
30-39	1110000
40-49	1111000
50-59	1111100
60-69	1111110
70-79	1111111

We applied categorical attributes into the model, indicating presence or absence of an attribute regarding the movie. These attributes assume binary values: zero means absence. One means presence. Table II shows this codification. Worth to mention that a movie may have more than one genre, for example, action and sci-fi. The approach is a reasonable solution since it activates the bits related to these genres.

TABLE II. Codification for categorical attributes.

Genre	Codification
Action	1000
Comedy	0100
Drama	0010
Sci-fi	0001

Different numbers of bits were dedicated to each attribute in the process of converting data into a binary sequence. Table III shows those numbers, which were empirically chosen. Hence, in the last phase, WiSARD receives a binary matrix processed in the previous phase. This binary matrix has patterns that is applied for training the model with different sizes of bits α. We tested a range of values and defined $\alpha = 120$. Finally, we calculated the overall accuracy.

TABLE III. Number of bits for each attribute.

Attributes	Number of Bits
Movie Genre	18
Year Movie	9
User Genre	18
User Age	12
Score Movie Average	28
Score User Average	14
Score User Average for Gender	21
Total	120

B. WiSARD processing

For training a discriminator which the following steps: the pattern to be trained is converted into binary, is then selected n bit random, where n is arbitrary and should be set empirically, the selected bits are used to enable one of its memory locations 2^n. Figure 3 illustrates the process for training a discriminator for a category I from a WiSARD with memories of 8 bits and input size of 12 bits. WiSARD tends to experience saturation problems. This problem is characterized by misclassifications when it learns with large amounts of data. Since several memory positions should be set to 1 due to a noisy observation [28]. Besides, WiSARD randomly choose a score when a tie occurs between the discriminators. To minimize the problem, a bleaching method [29] was applied.

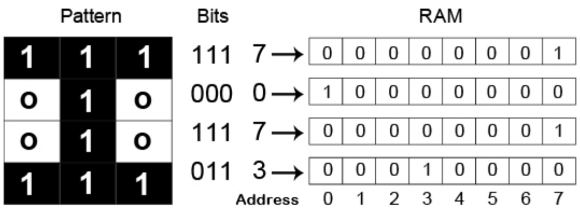


Fig. 3. Training the discriminator.

Bleaching method consists to store, in the WiSARD's memories, the number of times that a pattern was displayed for the recognition step. Thus, this deterministic response is applied as a tiebreaker and WiSARD results are improved. Besides, instead of store only boolean values into the memories, they also have values by unit, in an incremental process. However, the bleaching method has the possibility of increase the response time regarding prediction. Besides, it does not affect neither the complexity nor the training time.

Finally, the performance in test phase is analyzed regarding amount of data in the training phase, resulting in two tests: random data separation (TA) and balanced data separation (TB). Random data separation consists in randomly select the samples until reach a limit of instances to be trained. Balanced data separation consists in select the same quantity of samples from each category, resulting in a batch of samples divided equally. Thus, we intend to test and to choose the best approach among these two strategies, applying along with WiSARD and comparing with the baseline.

V. Experiments and Results

For this experiment we chose a dataset originated from MovieLens website [2]. This dataset was generated in April, 1998 and contains $100,000$ registers given by 943 users for $1,682$ movies. Each user has voted in at least 20 movies and these ratings are discrete numbers varying between 1 and 5, which 1 means a poor movie, and 5 an excellent movie. Discriminators represents movie ratings provided by the users.

A. Experiments

The IRSVD algorithm was chosen as a baseline for the experiments since this algorithm is vastly applied as an important baseline in several works, as mentioned earlier. Besides, were chosen three approaches for training the discriminators in WiSARD which we denominated them as WiSARD-1, WiSARD-2 and WiSARD-3:

- WiSARD-1: Five discriminators;
- WiSARD-2: Five discriminators for each genre;
- WiSARD-3: Five discriminators for each occupation.

[2]https://movielens.org/

Was applied the K-fold validation [30], with $K = 10$. K-fold validation consists in randomly split K parts of the original sample, commonly known as folds. Each fold in turn is applied for testing and the remaining folds are applied for training. Was executed the 10-fold experiment 10 times and was computed the final result based on the average.

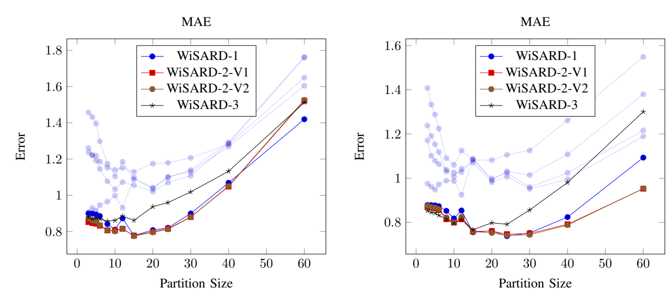
The evaluation metric was the mean absolute error (MAE) [31], that is, the average of absolute difference between the predictions and the actual labels. The experiment consists in:

- Separate data by genre;
- Train WiSARD discriminators with each genre;
- Search which n that produces the best MAE;
- Recommend items.

Since a movie can have more than one genre, were developed two versions of WiSARD-2, aiming to choose the best selecting criteria for a discriminator. The first version returns the discriminator who had the higher ratings given a genre, and was denominated WiSARD-2-V1. The second version returns the discriminator who had the best average results between the genres, and was denominated WiSARD-2-V2.

B. Results

Firstly, were tested the two approaches, TA and TB, in order to choose the best approach. After, was tested the best approach varying its partition size, in order to choose the best partition size for the problem. Figure 4 illustrates a comparison between TA (transparent) and TB (strongly colored) approaches. The input was divided in partition sizes of 2500 and 25000. Thus, were tested with a small partition size and a big one, aiming to verify the performance of the approaches. The quantity of memories chosen for the WiSARD in these experiments was 4.



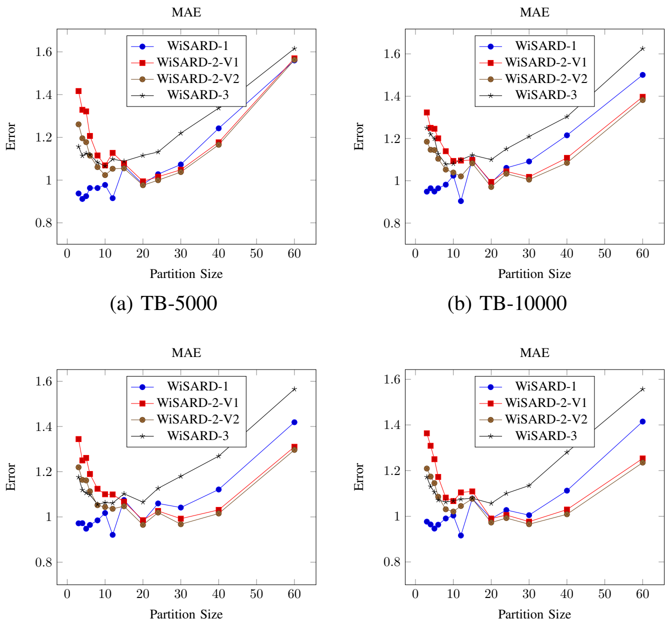
(a) TA-2500 with TB-2500

(b) TA-25000 with TB-25000

Fig. 4. TA approach versus TB approach.

The results showed that TB approach was faster and better than TA over the two different partition sizes. This indicate that an approach where we divide the genres in similar quantities increases the quality of the WiSARD predictions. In order to select the best partition size, we tested TB with intermediary partition sizes, ranging between 5000 and 20000. Thus, we can choose the best WiSARD approach for the best input approach (TB). The results are illustrated in Figure 5.

Until partition size = 12, WiSARD-1 approach had best errors results. However, the two variations of WiSARD-2 approach had better results than the others WiSARD approaches



(a) TB-5000 (b) TB-10000

(c) TB-15000 (d) TB-20000

Fig. 5. MAE results of all approaches regarding partition size.

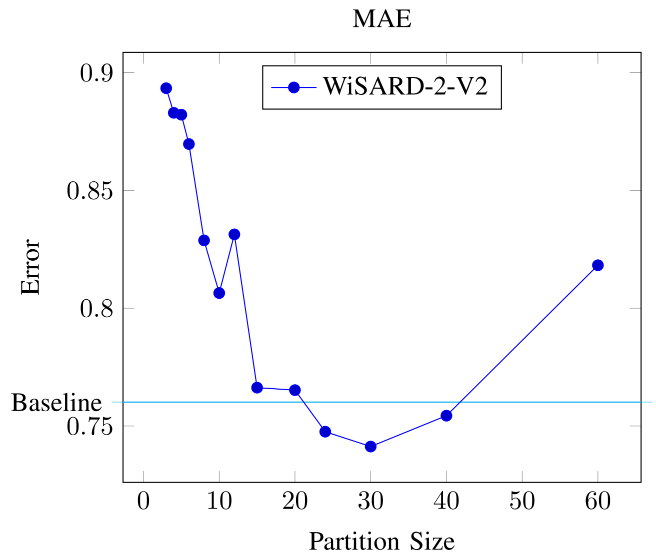
in the remaining partition sizes. Despite WiSARD-2 had worse results than WiSARD-1 in smaller partitions, the two versions of WiSARD-2 were stable over the partition sizes. These results indicate that the combination of a balanced dataset along with discriminators allocated for each genre increases the WiSARD results.

Worth to mention that, the WiSARD results are rather improved when we take into account the average results of these discriminators, as approached in WiSARD-2-V2. Therefore, WiSARD-2-V2 with TB approach was chosen. Finally, we applied the WiSARD-2-V2 with TB approach for 80% of the dataset for training and 20% for testing and compared with the baseline (blue line), as illustrated in Figure 6.

According to the results, WiSARD-2-V2 has better MAE results than IRSVD algorithm when the partition size varies between 22 and 42. Moreover, the best performance of WiSARD is achieved when using a partition size = 30, achieving a MAE = 0.7413 front of the IRSVD (MAE = 0.7702) an improvement of 3.7%, as shown in Figure 6, a parameter sensitivity table's regarding the partition sizes.

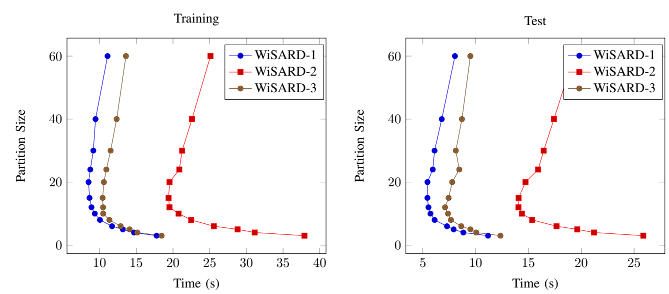
Regarding the runtime, Figure 7 illustrates the WiSARD runtime for the three approaches taking into account the partition size. Both train and test steps were analyzed. The methods WiSARD-1 and WiSARD-3 were faster than WiSARD-2, both for training and testing. Moreover, WiSARD-1 is the fastest approach. This is due to the low quantity of discriminators allocated for training and predicting steps.

Worth to mention that all methods were faster than the IRSVD algorithm. The training time of the IRSVD algorithm was 64.36 seconds whereas our proposal, WiSARD-2-V2 with TB approach, trained within 21 seconds. Thus, our proposal achieved 306% of improvement in speed.



(a) TA-80/20

Fig. 6. MAE results of the best WiSARD model regarding partition size.



(a) TA-80/20 (b) TA-80/20

Fig. 7. Relation between the partition size and the runtime for train and test.

VI. Conclusion

In this work, we proposed the use of weightless neural network WiSARD for predicting ratings on online recommender systems. We showed how we modeled the best partition sizes and data balancing through a three versions regarding the discriminators and two approaches regarding data input. The results indicated that our approach is 306% faster than IRSVD, a well-known state-of-the-art algorithm and 3.7% better than this baseline regarding the MAE.

A drawback of the algorithm is that depending on the quantity of model parameters, choose a reasonable quantity of bits for a particular parameter may be very difficult. Hence, a poor estimation of these parameters brings negative impact in the results. Therefore, is recommended to use optimization algorithms to select the best number of bits for each attribute.

As future work, we will test the WiSARD performance for large datasets. Another important future work is the use of the WiSARD for cold start problems in recommender systems. Worth to mention that the processing time of WiSARD can be improved, since the algorithm is easily parallelizable, including the use of GPU processors [32]. Thus, a future direction is to explore WiSARD in a parallelized or distributed architecture. Another interesting research direction is to consider that data may be non-stationary. Hence, the online model may perform special strategies to overcome the concept-drift problem [33].

References

[1] B. Schwartz, "The paradox of choice: Why more is less." Ecco New York, 2004.

[2] F. Ricci, L. Rokach, and B. Shapira, *Introduction to Recommender Systems Handbook*, 2015, vol. 54, no. OCTOBER. [Online]. Available: http://www.springerlink.com/index/10.1007/978-0-387-85820-3%5Cnhttp://link.springer.com/10.1007/978-0-387-85820-3_1

[3] G. Adomavicius and A. Tuzhilin, "Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions," *Knowledge and Data Engineering, IEEE Transactions on*, vol. 17, no. 6, pp. 734–749, 2005.

[4] J. L. Herlocker, "Understanding and improving automated collaborative filtering systems. 2000. 220 f," Ph.D. dissertation, Tese (Doutorado em Ciência da Computação)-University of Minnesota, Minnesota, 2000.

[5] J. B. Schafer, J. A. Konstan, and J. Riedl, "E-commerce recommendation applications," in *Applications of Data Mining to Electronic Commerce*. Springer, 2001, pp. 115–153.

[6] S. Chang, Y. Zhang, J. Tang, D. Yin, Y. Chang, M. A. Hasegawa-Johnson, and T. S. Huang, "Streaming recommender systems," in *Proceedings of the 26th International Conference on World Wide Web*. International World Wide Web Conferences Steering Committee, 2017, pp. 381–389.

[7] L. Song, C. Tekin, and M. van der Schaar, "Online learning in large-scale contextual recommender systems," *IEEE Transactions on Services Computing*, vol. 9, no. 3, pp. 433–445, 2016.

[8] H. J. Ahn, "A new similarity measure for collaborative filtering to alleviate the new user cold-starting problem," *Information Sciences*, vol. 178, no. 1, pp. 37–51, 2008.

[9] F. Cacheda, V. Carneiro, D. Fernández, and V. Formoso, "Comparison of collaborative filtering algorithms: Limitations of current techniques and proposals for scalable, high-performance recommender systems," *ACM Transactions on the Web (TWEB)*, vol. 5, no. 1, p. 2, 2011.

[10] Z. Huang, H. Chen, and D. Zeng, "Applying associative retrieval techniques to alleviate the sparsity problem in collaborative filtering," *ACM Transactions on Information Systems (TOIS)*, vol. 22, no. 1, pp. 116–142, 2004.

[11] I. Aleksander, M. De Gregorio, F. M. G. França, P. M. V. Lima, and H. Morton, "A brief introduction to weightless neural systems." in *ESANN*. Citeseer, 2009, pp. 299–305.

[12] M. C. W. Keat, R. Abdullah, R. A. Salam, and A. A. Latif, "Weightless neural network array for protein classification," in *Parallel and Distributed Computing: Applications and Technologies*. Springer, 2004, pp. 168–171.

[13] D. Cardoso, M. De Gregorio, P. Lima, J. Gama, and F. França, "A weightless neural network-based approach for stream data clustering," in *International Conference on Intelligent Data Engineering and Automated Learning*. Springer, 2012, pp. 328–335.

[14] M. De Gregorio and M. Giordano, "Change detection with weightless neural networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2014, pp. 403–407.

[15] M. Berger, A. F. De Souza, J. de Oliveira Neto, E. de Aguiar, and T. Oliveira-Santos, "Visual tracking with vg-ram weightless neural networks," *Neurocomputing*, vol. 183, pp. 90–105, 2016.

[16] M. De Gregorio and M. Giordano, "Background estimation by weightless neural networks," *Pattern Recognition Letters*, vol. 96, pp. 55–65, 2017.

[17] J. C. M. Oliveira, K. V. Pontes, I. Sartori, and M. Embiruçu, "Fault detection and diagnosis in dynamic systems using weightless neural networks," *Expert Systems with Applications*, 2017.

[18] D. O. Cardoso, J. Gama, and F. M. França, "Weightless neural networks for open set recognition," *Machine Learning*, vol. 106, no. 9-10, pp. 1547–1567, 2017.

[19] I. Aleksander, W. Thomas, and P. Bowden, "Wisard: a radical step forward in image recognition," *Sensor review*, vol. 4, no. 3, pp. 120–124, 1984.

[20] T. B. Ludermir and W. R. de Oliveira, "Weightless neural models," *Computer standards & interfaces*, vol. 16, no. 3, pp. 253–263, 1994.

[21] P. Resnick, N. Iacovou, M. Suchak, P. Bergstrom, and J. Riedl, "Grouplens: an open architecture for collaborative filtering of netnews," in *Proceedings of the 1994 ACM conference on Computer supported cooperative work*. ACM, 1994, pp. 175–186.

[22] R. J. Hunt, "Percent agreement, pearson's correlation, and kappa as measures of inter-examiner reliability," *Journal of Dental Research*, vol. 65, no. 2, pp. 128–130, 1986.

[23] A. Paterek, "Improving regularized singular value decomposition for collaborative filtering," in *Proceedings of KDD cup and workshop*, vol. 2007, 2007, pp. 5–8.

[24] Y. Koren, R. Bell, and C. Volinsky, "Matrix factorization techniques for recommender systems," *Computer*, no. 8, pp. 30–37, 2009.

[25] S. A. Dudani, "The distance-weighted k-nearest-neighbor rule," *IEEE Transactions on Systems, Man, and Cybernetics*, no. 4, pp. 325–327, 1976.

[26] M. Norouzi, D. J. Fleet, and R. R. Salakhutdinov, "Hamming distance metric learning," in *Advances in neural information processing systems*, 2012, pp. 1061–1069.

[27] A. Kappaun, K. Camargo, F. Rangel, F. Firmino, P. M. V. Lima, and J. Oliveira, "Evaluating binary encoding techniques for wisard," in *Intelligent Systems (BRACIS), 2016 5th Brazilian Conference on*. IEEE, 2016, pp. 103–108.

[28] H. C. de Castro Carneiro, "Theoretical results on a weightless neural classifier and application to computational linguistics," Ph.D. dissertation, Universidade Federal do Rio de Janeiro, 2017.

[29] D. S. Carvalho, H. C. Carneiro, F. M. França, and P. M. Lima, "B-bleaching: Agile overtraining avoidance in the wisard weightless neural classifier." in *ESANN*, 2013.

[30] R. Kohavi *et al.*, "A study of cross-validation and bootstrap for accuracy estimation and model selection," in *Ijcai*, vol. 14, no. 2. Montreal, Canada, 1995, pp. 1137–1145.

[31] K. Goldberg, T. Roeder, D. Gupta, and C. Perkins, "Eigentaste: A constant time collaborative filtering algorithm," *Information Retrieval*, vol. 4, no. 2, pp. 133–151, 2001.

[32] A. J. da Silva, W. R. de Oliveira, and T. B. Ludermir, "Training a classical weightless neural network in a quantum computer," *Quantum*, vol. 1, no. 1, p. 1, 2014.

[33] R. S. Ferreira, G. Zimbrão, and L. G. M. Alvim, "Amanda: Density-based adaptive model for non-stationary data under extreme verification latency scenarios," Ph.D. dissertation, Universidade Federal do Rio de Janeiro (UFRJ), 2018.