Context-Aware Recommendations using Differential Context Weighting and Metaheuristics

Kunal Gusain1 and Aditya Gupta1

1 Computer Science Department, Bharati Vidyapeeth’s College of Engineering, Guru Gobind Singh Indraprastha University, Delhi, India

[kunalgusain1995@gmail.com and](mailto:kunalgusain1995@gmail.com%20and) adityag95@gmail.com

**Abstract.** Context plays a paramount role in language and conversations, and since their incorporation into traditional recommendation engines, which made use of just the User and Item details, an effective method to utilise them in the best possible manner is of great importance. In this paper, we propose a novel approach to handle the sparsity of contextual data, their increasing dimensionality and the development of an effective model for a Context-Aware Recommender System (CARS). We further go on give relevance, in the form of assigning weights even to the individual attributes of each context. Differential Context Weighting (DCW) is used as the rating model to obtain the desired ratings. Optimisation of weights required for DCW is done through metaheuristic techniques, and towards this, we have further gone on to experimentally compare two of the most popular ones, namely Particle Swarm Optimization (PSO) and the Firefly Algorithm (FA). Recommendations using the optimal one were then obtained.

**Keywords:** recommender system; context; context-aware recommendation; differential context; metaheuristics; particle swarm optimization; firefly algorithm;

1. Introduction

Recommender systems continue to be one of the most researched fields in recent times and find extensive usage in e-commerce, personalized marketing or recommending products, places, music amongst other things. An internet user is quite likely to come across a recommendation service, sometimes without even being explicitly aware of it. Whereas one cannot deny the advantage a good and timely recommendation offers to the user, we do however have to accept that a huge margin of improvement still remains in their working. Usage of a high number of, and forever evolving, artificial intelligence and machine learning techniques, makes sure that there is a boundless potential in terms of their optimization. Traditional recommendation engines were used to working on just two factors, which are the Users and the Items. More recently there have been efforts to include “contexts” in the recommendation process as well [1], enabling us to give even more specific and improved suggestions. These recommendation engines are called Context-Aware Recommender Systems or simply CARS.

Context, one entity that has the potential to alter, or put into perspective any conversation or communication we have. Over time its role in recommendation engines has also grown, helping to provide more potent and personalised suggestions. A context can be as simple as the users’ location, day, time, or it could be their mood and even their companions. Both implicit and explicit contexts are relevant, and as contexts do, they have the ability to change over time for the same user. Context-Aware Recommender Systems (CARS) incorporate them in the hope to provide efficient recommendations, and towards this extent, a number of models have been put forward. There are two primary ways in which this process of incorporation takes place, namely the Filtering and the Modelling approach as given by Adomavicius et al. [2]. The filtering approach is further subdivided into two approaches, depending on the time filtering is done, namely pre-filtering and post-filtering approach. However, in both of these the contexts are not incorporated directly instead used as a filter, thus the second approach of modelling is preferred since it directly utilizes contexts to give recommendations. Once the approach has been finalized we now have to choose a rating model to find the desired ratings.

One rating model that could be used in CARS is the Differential Context Relaxation (DCR) model [3, 4]. In DCR a subset of features is selected that are then used for giving recommendations. What happens as a result is that not all features are effectively incorporated and instead the suggestions are just based on a small subset of, sometimes even arbitrary constraints. Due to the lack of an effective model, it was perhaps necessary at the beginning to tackle the problem of sparsity, but now the more nuanced and accurate approach call Differential Context Weighting (DCW) has stormed the marketplace of ideas and has been shown to be a substantial improvement over DCR. This pioneering work was also done by a team of the foremost researchers and authorities in the field of contextual recommendations, Zheng et at. [5]. DCW started giving weights to each of the features, and then these weighted contexts were used for evaluation. So unlike in the relaxation technique of leaving behind some contexts, now all the features could be included and the end recommendation was as a result more potent. The process of assigning weights further needs an optimization step, wherein we decide how and what are the weights that are going to be assigned to the features. This optimization can be done by applying metaheuristic techniques.

Metaheuristics [6, 7] are being increasingly used and researched in optimization related methodologies, primarily due to their ability to reduce the time in computation and also since they are independent of gradient based analysis; the caveat is that a globally optimal solution is not necessarily always found. Our focus is on nature-based metaheuristic techniques, namely Particle Swarm Optimization (PSO) and Firefly Algorithms (FA). PSO, like other swarm intelligence techniques, is inspired by insects, birds, or animals’ behaviour in nature and how they swarm [8, 9]. A population-based optimization, where each particle has a defined position in the space of particles, and depending upon the velocity, which is the speed and direction, their convergence can be evaluated. Firefly Algorithms, unlike PSO which used bees and birds as a reference, uses fireflies and their organization as a metric [6, 10, 11], with the idea being the correlation between the brightness/glow of the fireflies to the values of the objective function. In our paper, we have compared and analysed both these methods across a number of metrics to find out which would be more suitable for optimization in DCW, and eventually better recommendations in CARS.

The objective of this paper is to find an efficient and economic approach of giving context-aware recommendations. Towards this extent, we have used DCW as a rating model. We have further enhanced and improved upon the existing methodology of using contexts by giving relevance in the form of weights to the individual attributes of all the features. Let us say that if the choice of companion is a feature, then apart from giving weights to the entire metric, individual attributes of a companion like family, alone, significant other and son, have also been assigned weights. This has helped to make the recommendations even more potent, and to the best of our knowledge, this method hasn’t been implemented before. The data we use, which is of movie ratings, is vast and has a plethora of features to choose upon from. Due to the paucity of good contextual data available, we have used one of the most extensive data available, and the permission for the same has also been obtained. We have further gone on to compare two of the most usable, popular and efficient optimization techniques that exist, PSO and FA, and compared them on a number of metrics to find which of them is more suitable to our problem. PSO was found to be the better alternate and was thus used to obtain desired recommendations.

1. Differential Context Weighting (DCW)

Differential Context Relaxation (DCR) [3] works on the principle of assigning binary values to each context i.e. 0 or 1. A value of ‘1’ when the context is present and ‘0’ when the context is absent. Unlike DCR, Differential Context Weighting (DCW) works on the principle of assigning some weights to the given contexts instead of just assigning them values of ‘0‘or ‘1’ [5]. These weights have the range of all real numbers between ‘0’ and ‘1’. These weights can be seen as the amount of contribution each context makes to the final rating. If a context has more weight, then that means it contributes more to the actual rating as compared to the context with low weights. The similarity metric which we use for calculating the similarity between two users, given the weights for each context is weighted, is Jaccard similarity function. The key parameters for the weighted Jaccard similarity function are σ sigma and ‘c, d’, where sigma is the vector containing weights for all the contexts, whereas c and d are the given contexts for the user. The weighted Jaccard similarity, which is used to calculate the similarity between all the users is given by equation (1):

(1)

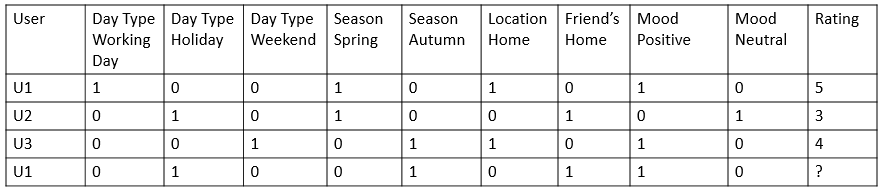
An example of the dataset used for movie recommendations is given in Table 1.

**Table 1.** Example: contextual ratings of users for movie *‘i’*



A significant change that we have proposed is the assignment of weights to the attributes of the individual context. This makes the dataset even sparser. This can be done by first converting the data in Table 1. To the bit matrix like the one shown in Table 2.

**Table 2.** Bit matrix of Table 1



The DCW model is given in Equation (2):

 (2)

‘P’ is the predicted rating, ‘a’ is the user, ‘i’ is the item for which the rating ‘r’ has been assigned with being the rating function and  is the baseline function.

The equation mainly consists of 4 parts:

1. Selection of Neighbours
2. Contribution by Neighbours
3. User Baseline
4. Similarity between users

Selection of neighbours involves selecting all the users from the data who have rated the given item ‘i’ in the given context ‘c’. It is also possible that a neighbour ‘u’ might have rated the movie differently in different context, so to tackle this problem we take maximally similar context of the neighbour and use it for the rating. The equation for selection of neighbours is given in (3).

(3)

Neighbour contribution is the process of calculating the contribution by the user to the rating of the user. Sometimes it is possible that the neighbour has rated the same item in multiple contexts differently. So we take the weighted average of the rating given by that neighbour in all the contexts as given by equation (4).

(4)

Equation (5) shows the formula of overall average for all those items that were given ratings for similar contexts, ‘Iu’ depicts the set of all the items that were rated by our user ‘u’.

(5)

User baseline is the average rating given by that user in that similar context.

1. Metaheuristics

Recommendation engines, like life, require continuous concerted efforts to arrive at a feasible and desired result. Thus the heuristic approach is built into the fundamental structure of recommendation systems, and this trial and error as well as the continuous improvement structure of theirs makes metaheuristics indispensable in optimization techniques. A number of metaheuristic algorithms are available [12, 13] and their efficiency for solving problems helps them stay relevant in optimization domains. Out of all the population-based methods or evolutionary algorithms, Particle Swarm Optimization (PSO) and Firefly Algorithm (FA) are the most relevant ones pertaining to our problem set.

* 1. 3.1 Particle Swarm Optimization (PSO)

This swarm intelligence technique takes inspiration from the behaviour of swarming creatures like bees and organizational patterns of birds. PSO [9] works by identifying a set of target points, where all particles are plotted in the target space. Once all the target points are obtained, a concerted effort is undertaken in which at each successive step PSO tries to improve the given solution, in an effort to find the value which is closest to the target value. Ascribing velocity and direction to particles, it primarily has three components, namely momentum, cognitive and the social component. The first one is used to describe the earlier velocity, due to which the particle was able to reach the current position. Cognition is that property of the particle which makes it want to return to the best possible position/scenario it had come across during its travails to the current position. Finally, the third and the social component, another beautiful feature of this algorithm, describes the tendency of the particle to attach itself or move towards the best possible position in its neighbouring target space.

PSO has a few clear and distinct advantages, which makes it one of the more used and researched optimization algorithms. First is its tendency to be aware and knowledgeable about its environment and the population of particles in the target space that it is working in. Secondly, it has a fast convergence characteristic, which makes it time efficient. Finally, PSO has shown to be highly efficacious in obtaining results in static as well as dynamic search spaces

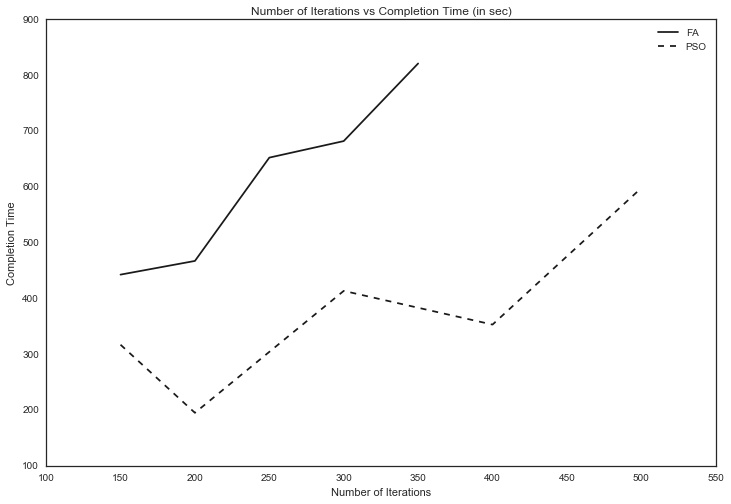
* 1. 3.2 Firefly Algorithm (FA)

Firefly Algorithm (FA) [6, 10, 11], another of the metaheuristic approaches that can be utilized in solving optimization problems. One of the main features that brought FA into the limelight was its ability to find optimal solutions even with a low number of iterations, and that too whilst maintain a high convergence rate. Also a swarm intelligence technique, FA works by studying the behavioural pattern of fireflies and how they mate with each other. Glowing in the dark, these bioluminescent creatures make use of their light to attract mates to breed. In the search space, one firefly is used to represent a single candidate. Using the brightness as a metric, these fireflies are then evaluated on their ability to absorb and emit light, as well their inherent attractiveness. Light intensity is the direct measurement of how attractive a firefly is. These characteristics of FA are possible because of certain natural features which make them rather unique and interesting: First of all, fireflies are unisex in gender, and attract others only by the strength or intensity of their glow. Secondly, it also has the ability to move randomly in search of a brighter prospect. Lastly, the degree of the glow of a firefly decreases in proportion to the decreasing distance between them, as the two creatures move towards one another. One of the basic features of any metaheuristic algorithms is its ability to randomize the paths occasionally, such that it allows us to avoid local optimums. In FA as well, the randomizing of path factor plays a similarly crucial role. After each move of the firefly, its brightness function is updated by evaluating it at this recent position. This brightness function is nothing but a representative manifestation of the objective function. At each successive move, the function is evaluated, and if it is found to be better, it is updated at this new location. We set limits to the possible movements, by deciding for how many iterations it can be run, or we can also have a cut-off value, which once reached results in the termination of the algorithm. A number of variations and hybrids have been proposed over time, and many trade-offs have also been discussed, however, the basic idea remains same, that of models being made based on the behaviours of bioluminescent fireflies, their mating habits and their attractiveness to each other being a characteristic of the degree of their brightness. All these qualities have made them a fascinating phenomenon and their effectiveness has caused us to study it as a problem-solving alternate.

1. Experimental Setup

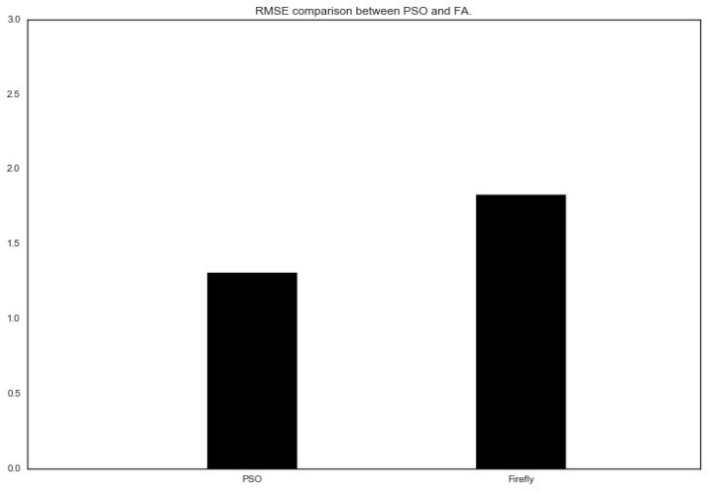
We use the data set of a movies recommendation engine. With more than a thousand movies, the data has around fourteen different contexts like the users age, their sex, city, country, time of the day, location, weather, companion, mood, emotional state, interaction (whether it is the first interaction with a movie or n-th time) and so on. The attributes of these contexts are also quite rich like the emotional state has Sad, happy, Scared, Surprised, Angry, Disgusted as its attributes. Since in our research proposal we want to treat each attribute as equally important and contributing feature like individual contexts, thus we have a total of seventy-four features, for which we need to calculate the recommendation on. This LDOS – CoMoDa, movie dataset is a sparse one. We tested our results, or obtained Root Mean Squared Value (RMSE) for over a thousand entries.

Pertaining to our first proposal, which was to assign weights even to the individual attributes of the contexts in the dataset. Such that fourteen or so contexts now turned into over seventy-four features. This though made the data sparse, it also made the recommendations in the end more potent with a reduced RMSE, since now even more user and context specific recommendations were being obtained. Now we have the newly expanded seventy-four features, which need to be optimized and assigned weights before being entered into our rating model. Since there are a number of optimization techniques available, our research has tried to narrow down and find the most effective technique that could be used for the aforementioned purpose. We use the, extremely popular and widely used PSO as the first technique, and FA as the second one, since both of these offer the most relevant methodologies that can be applied to the said problem. The results of the optimization techniques would be the optimized weights for each context, to be entered into the rating model. (DCW here).



**Fig. 1.** Number of Iterations vs Completion Time graph for PSO and FA.

The first metric for comparison is the Number of Iterations vs Completion Time (in seconds), given in Fig. 1. Number of iterations for both the algorithms, PSO and FA were compared over a wide margin to gauge their behaviour and offer a succinct analysis. We observed that with an increase in the number of iterations, there was a gradual but sure increase in the completion times of both the algorithms. However, both also showed some erratic behaviour with the iteration increase, which is characteristic of all metaheuristic algorithms due to their inherent random functions, thus does not significantly deter our prognosis. As is evident from Fig. 1. PSO is significantly better than FA for our movie recommendation data.



**Fig. 2.** RMSE comparisons for PSO and FA.

The second metric used was to find their accuracy in giving recommendations, this was done by finding and comparing their RMSE values. The graph for the comparison is given in Fig. 2. This again showed PSO at an advantage with significantly less RMSE for the given data, PSO had 1.3079 whereas FA had 1.8288. Thus the two metrics employed by us, helped to evaluate and find out that PSO is markedly better that its alternate FA.

The final step is the implementation of the rating model using the weighted contexts obtained from the metaheuristic optimization techniques used above. The model we use is the Differential Context Weighting or the DCW model. For finding the similarity between the users in DCW, we use the Weighted Jaccard similarity function given in equation (1). Also, the ratings were calculated by using the DCW equation given in (2). Thus we were able to successfully implement a context-aware recommendation engine incorporating our proposed suggestions.

1. Conclusion and Future Scope

The aim of this paper has been to find and implement an effective, economic and efficacious Context-Aware Recommendation System (CARS), and a humble effort to further the research in how contexts can be further utilized to give desired and accurate recommendations. We improved upon the utilization and inclusion of contexts in CARS by assigning individual weights, to each of the features’ attributes. For example, if the mood of the user was one of the contexts, then neutral, positive and negative each has been given individual values, thus increasing their significance in giving recommendations. To the best of our knowledge, evaluation, and usage of individual contexts has not been done before. The eventual, extremely low Root Mean Squared Error or RMSE value offers confirmation that their inclusion helps improve the results. CARS first needs weighted contexts to be used as an input in the rating model (DCW here). Thus, we analysed and compared two of the most popular and relevant optimization techniques to be used to provide weights to the contexts. These techniques were the metaheuristic, PSO, and FA. Experimental study proved that PSO was far more suited for contextual recommendations. It outperformed FA by a fair estimate and helped make the recommendation system even better. Once these values from metaheuristic approaches were obtained, using Jaccard Similarity, the similarity between different users was found. Here we take DCW and not the relaxation technique DCR since this is a more accurate and efficient methodology [5]. DCW also been shown to be superior to other rating models as well [3, 4, 14]. Thus using these series of steps, optimal final recommendations were arrived at and they had an extremely low RMSE value, thus giving us an efficient CARS.

References

1. D.B. , Schilit Theimer , C.A. Brunk , C. Evans , B. Gladish , M. Pazzani , Adaptive interfaces for ubiquitous web access, Comm. ACM 45 (5) (2002) 34–38 .
2. G. Adomavicius, T. Alexander, Context-aware recommender systems, in Recommender Systems Handbook, Springer, US, 2011, pp. 217–253.
3. Zheng, Y., Burke, R., Mobasher, B.: Differential context relaxation for context-aware travel recommendation. In: Huemer, C., Lops, P. (eds.) EC-Web 2012. LNBIP, vol. 123, pp. 88–99. Springer, Heidelberg (2012).
4. Zheng, Y., Burke, R., Mobasher, B.: Optimal feature selection for context-aware recommendation using differential relaxation. In: ACM RecSys 2012, Proceedings of the 4th International Workshop on Context-Aware Recommender Systems (CARS 2012). ACM (2012).
5. Zheng, Yong, Robin Burke, and Bamshad Mobasher. "Recommendation with differential context weighting." International Conference on User Modeling, Adaptation, and Personalization. Springer Berlin Heidelberg, 2013.
6. X. S. Yang, “Nature-Inspired Metaheuristic Algorithms”, Luniver Press, 2008.
7. Christian Blum, Maria Jos´e Blesa Aguilera, Andrea Roli, Michael Sampels, Hybrid Metaheuristics, An Emerging Approach to Optimization, Springer, 2008.
8. Xiang-yin Meng, Yu-long Hu, Yuan-hang Hou, Wen-quan Wang, The Analysis of Chaotic Particle Swarm Optimization and the Application in Preliminary Design of Ship”, International Conference on Mechatronics and Automation, August, 2010.
9. J. Kennedy, R. C. Eberhart, “Particle swarm optimization”, IEEE International Conference on Neural Networks, Piscataway, NJ., pp.942-1948, 1995.
10. Sh. M. Farahani, A. A. Abshouri, B. Nasiri, and M. R. Meybodi, “A Gaussian Firefly Algorithm”, International Journal of Machine Learning and Computing, Vol. 1, No. December 2011.
11. Xin-She Yang, Chaos-Enhanced Firefly Algorithm with Automatic Parameter Tuning, International Journal of Swarm Intelligence Research, December 2011.
12. P. M. Pardalos and H. E. Romeijn, Eds., Handbook of Global Optimization, vol. 2, Kluwer Academic Publishers, 2002.
13. S. Luke, Essentials of Metaheuristics, Lulu, 2009
14. Zheng, Yong, Bamshad Mobasher, and Robin Burke. "Integrating context similarity with sparse linear recommendation model." International Conference on User Modeling, Adaptation, and Personalization. Springer International Publishing, 2015.
15. Unger, Moshe, et al. "Towards latent context-aware recommendation systems." Knowledge-Based Systems 104 (2016): 165-178.