Analysis of the recommendation algorithm in COHESY

Igor Kulev[1], Elena Vlahu-Gjorgievska[2], Saso Koceski[3], Verica Bakeva[1]

[1]Faculty of Computer Science and Engineering, University "Ss Cyril and Methodious", Skopje, Macedonia
{igor.kulev, verica.bakeva}@finki.ukim.mk
[2]Faculty of administration and information systems management, University "St.Kliment Ohridski", Bitola, Macedonia
elena.vlahu@uklo.edu.mk
[3]Faculty of Computer Science, University "Goce Delcev", Stip, Macedonia
saso.koceski@ugd.edu.mk

Abstract. Pervasive health care takes steps to design, develop, and evaluate computer technologies that help citizens participate more closely in their own healthcare, on one hand, and on the other to provide flexibility in the life of patient who lead an active everyday life with work, family and friends. This paper presents a novel collaborative algorithm that generates recommendations and suggestions for preventive intervention. The main purpose of this algorithm is to find the dependency of the users' health condition and physical activities he/she performs. The recommendation algorithm, presented in this paper, is part of the Collaborative health care system model called COHESY. COHESY improves quality of care and life to its users, by offering freedom to enjoy life with the confidence that a medical professional is monitoring theirs health condition.

Keywords. Personal healthcare, recommendation algorithm, classification, clustering.

1 Introduction

Providing patients with convenient health facilities at a low cost has always been a great challenge for health service providers. Moreover, the fast changing life style of the modern world and the problem of aging society pose an urgent need to modernize such facilities. This involves devising cheaper and smarter ways of providing healthcare to disease sufferers. In addition, emphasis has to be paid on providing health monitoring in out-of-hospital conditions for elderly people and patients who require regular supervision, particularly in remote areas. Future trends in national healthcare services are expected to include shorter hospital stays and better community care.

Patient-centered development process is useful for healthcare information system in order to reduce system complexity and increase the usability [1]. Pervasive health care takes steps to design, develop, and evaluate computer technologies that help citizens participate more closely in their own healthcare [2], on one hand, and on the

other to provide flexibility in the life of patient who lead an active everyday life with work, family and friends [3]. However, these systems do not consider collaborative value that can be provided with matching gathered data.

The collaborative health care system model, called COHESY, gives a new dimension in the usage of novel technologies in the healthcare. This system uses mobile, web and broadband technologies, so the citizens have ubiquity of support services where ever they may be, rather than becoming bound to their homes or health centers. The most important benefits of COHESY are possibility for patient notification in different scenarios, transmissions of the collected biosignals (health parameters as blood pressure, heart rate) automatically to medical personnel and increased flexibility in collecting medical data. But the main components and advantages of COHESY, which differentiates it from other health care systems, are the usage of the social network and its' recommendation algorithm.

The social network allows connecting users with same or similar diagnoses, sharing their results and exchanging their opinions about performed activities and received therapy. At the same time, collaborative algorithms generate average values based on filtering large amounts of data about concrete conditions as are geographical region, age, sex, diagnosis, etc. In this way, recommendation algorithm gives recommendations to the users for performing a specific activity that will improve their health. These recommendations are based on the users' health condition, prior knowledge derived from users' health history, and the knowledge derived from the medical histories of users with similar characteristics.

Users need to be provided with instant feedback and this is why the system performance is very important. The algorithms implemented in COHESY are efficient and are designed to have as low complexity as possible. These algorithms are also flexible and can easily be adapted to deal with different problem variations. There is also a possibility of generating more specific recommendations by exploring the information provided with each activity.

2 COHESY – Collaborative Health Care System Model

The collaborative health care system model COHESY gives a new dimension in the usage of novel technologies in the healthcare. This system model uses mobile, web and broadband technologies, so the citizens have ubiquity of support services where ever they may be, rather than becoming bound to their homes or health centers. Broadband mobile technology provides movements of electronic care environment easily between locations and internet-based storage of data and allows moving location of support. The use of a social network allows communication between users with same or similar condition and exchange of their experiences.

COHESY has simple graphical interfaces that provide easy use and access not only for the young, but also for elderly users. It has more purposes and includes use by multiple categories of users (patients with different diagnoses). Some of its advantages are scalability and ability of data information storing when communication link

fails. This model is interoperable system that allows data share between different systems and databases.

COHESY is deployed over three basic usage layers. The first layer consists of the bionetwork (implemented from various body sensors) and a mobile application that collects users' bio data and parameters of physical activities (e.g. walking, running, cycling). The second layer is presented by the social network which enables different collaboration within the end user community. The third layer enables interoperability with the primary/secondary health care information systems which can be implemented in the clinical centers and different policy maker institutions. The data information in this system are: users' personal data (name, age, height, diagnosis, therapy), data from users' bionetwork (weight, heart rate, blood pressure, blood-sugar level), realized and recommended activity (type of activity, path length, time interval, average speed), weather conditions, recommendation and suggestions. Different data information are exchanged between different layers of COHESY.

COHESY is an infrastructure that enables various personal healthcare scenarios. For example, it enables matching of performed user activity, by combining various data, including: length of path crossed, duration, speed of movement, medical condition of the user (heart rate, blood pressure - before and after , blood sugar level - before and after performed activity), weather conditions (atmospheric pressure, humidity, temperature), what is the medical diagnosis or therapy of the user (if there are any) and it can generate recommendation when certain patient should perform walk, with what pace and duration.

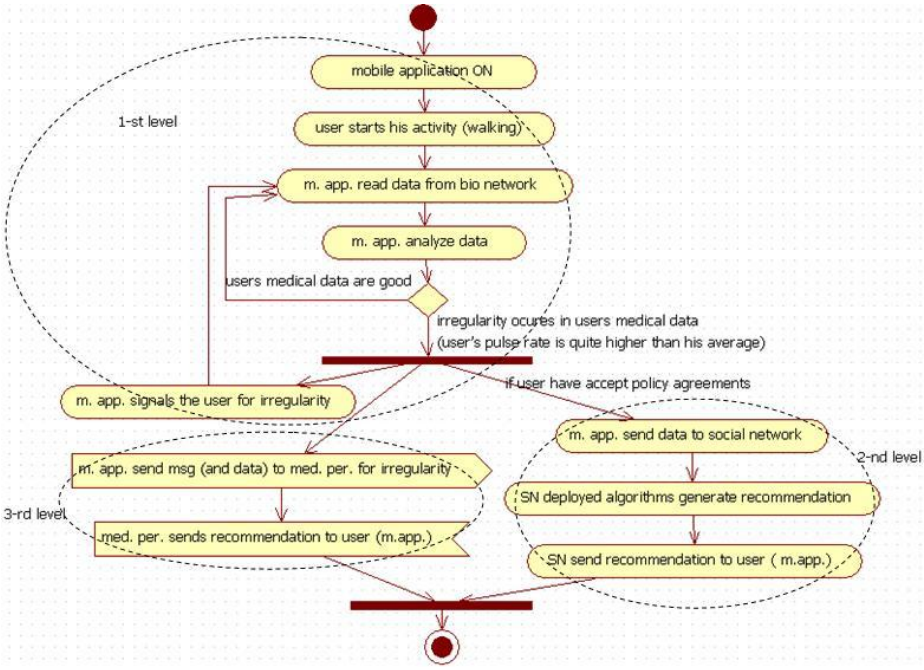

Fig. 1. Activity diagram for proposed scenario

One possible scenario is presented in Fig.1. The user (with diagnosed diabetes) switches on the application on his phone. Application using Bluetooth connects to the device that measures user weight, blood pressure and blood-sugar level and reads the measured parameters. Application sends recommendation to the user, generated by algorithms deployed on the social network, with activity which is best for his health to be performed. User starts his activity (running). The application reads the users' heart rate from his/her bionetwork. During the running an irregularity occurs. While reading the data, application detects that user's heart rate is quite higher than his/her average and the application sends message with those data to the medical center, social network (if patient has already agreed to security and privacy statements of the social network) and signals the user that there is some irregularity happening.

Medical personnel can review the submitted data and previous user's medical records. Based on the user's diagnosis, treatment received and his activity currently carried out, along with the medical data received from the application, medical personnel decides that the user should stop running and take pause for 15 minutes. This recommendation is issued back to the application of the user. The application signals to the user that a message from the medical center has arrived. The user applies the recommendation from the medical center. The same recommendation can be generated by algorithms deployed on the social network that are based on the average data and previously generated successful recommendations from the social network, previous clinically originated recommendations and patient history. These two recommendations differ in the validity as explained in [4].

3 Recommendation systems

Recommendation systems are used extensively by sites which want to give users better experience and they do this by giving suggestions to users about items they may like. But before giving the recommendation, the site first needs to learn user's preferences and it does that by examining the items he already said that he liked or by directly asking the user about his preferences.

There are few definitions for recommendation systems. According to the authors of [5] "Recommender Systems are software tools and techniques providing suggestions for items to be of use to a user. The suggestions provided are aimed at supporting their users in various decision-making processes, such as what items to buy, what music to listen, or what news to read."

Recommendation systems are usually classified into three categories, based on how recommendations are made: content-based filtering, collaborative filtering and hybrid techniques [6]. In content-based recommendation, the system tries to recommend items similar to those a given user has liked in the past, whereas in collaborative recommendation, the system identifies users whose tastes are similar to those of the given user and recommends items they have liked [7]. Hybrid system can incorporate the advantages of both methods while inheriting the disadvantages of neither.

Before giving any recommendations, the system must learn the user preferences. This is done by gathering and analyzing data about user's interaction with the system.

Explicit user data includes favoring or ranking an item while implicit data includes the viewing time, number of views, and actual purchases of an item.

Content-based recommendation systems analyze item descriptions to identify items that are of particular interest to the user [8]. These systems can learn user preferences from the set of items that the user liked in the past [5]. The items are characterized with values for different attributes. Also, the user preferences can be represented with desired values for each attribute. The system recommends to the user a set of items which are closest match to his desired values. Content-based systems can give recommendations by knowing the user preferences only. However, when implementing a content-based system, often it is necessary to have experts from the domain that would analyze the items, or that would provide the expertise to implement an automated process for item evaluation (machine learning algorithms can be used in this part of the system).

Collaborative filtering recommendation systems produce user specific recommendations of items based on patterns of ratings or usage (e.g., purchases) without knowing the features of the items. These systems calculate similarity between two users by analyzing the set of items that are liked by both users. This technique first finds the set of similar users to given user and then recommends items that are liked by most of the similar users. This is different from the content-based system because collaborative filtering systems treat the items as "black boxes", and the other systems examine the content. Collaborative filtering systems need to have data from many users in order to give better recommendations.

Evaluating recommendation systems is very difficult, because different algorithms may be better or worse on different datasets [9]. Users want as better recommendations as possible but recent researches show that when each algorithm is tuned to its optimum, they all produce similar measures of quality – there is a "magic barrier" where natural variability may prevent us from getting much more accurate.

4 Recommendation algorithm in COHESY

The recommendation algorithm is part of the second level, the social network, in COHESY. The main purpose of this algorithm is to find the dependency of the users' health condition and physical activities they perform. The algorithm incorporates collaboration and classification techniques in order to generate recommendations and suggestions for the physical activities that the users should carry out in order to improve their health. To achieve this we consider datasets from the health history of users and use classification algorithms on these datasets for grouping the users based on their similarity.

Although our recommendation system is not very similar with the most popular recommendation systems used in different contexts, we can still make analogy between the main concepts introduced in the previous section. In our context, we talk about physical activities and their influences on the change of the health parameters instead of talking about items and their attributes.

4.1 Levels of filtering

Our algorithm uses three levels of filtering, as shown in Fig.2. The first step is classification. All users belong to some diagnosis class (normal diabetes, heart problems). All users with different diagnosis from the diagnosis of the given user are filtered out. This step is important because some activities may be harmful for a particular group of people e.g. running may have much different effect on people with heart problems as opposed to people which are physically active.

The second level of our recommendation algorithm is the collaborative filtering. Every user has its own history of health conditions (health profiles) and it is important to find similar users to the given user which at some point of time in the past had similar health condition to the health condition of the given user at the moment. The technique that is used here can be considered as a collaborative filtering technique where items are equal to health profiles.

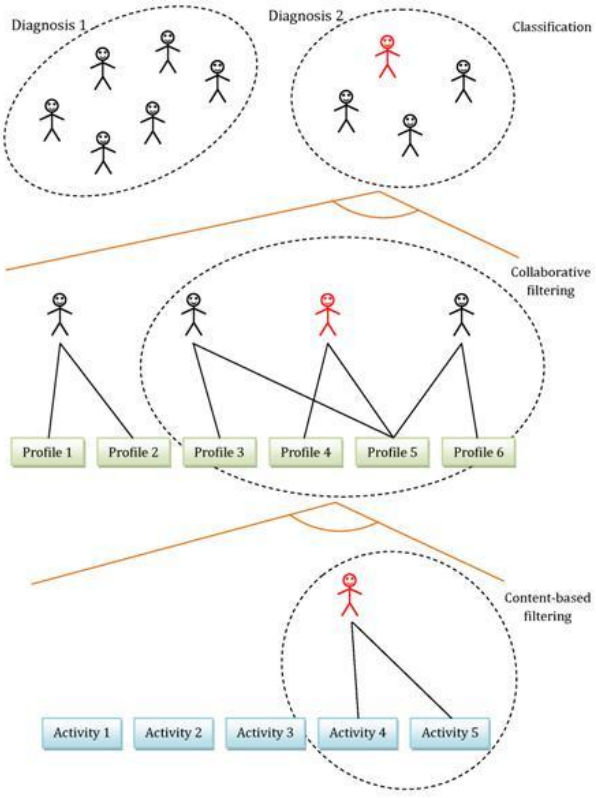

Fig. 2. Levels of filtering in recommendation algorithm

When the similar users are chosen, we use all their health condition history and history of performed activities to find the influences of each activity on the change of

the health parameters. Now we come with a fairly good approximation of the potential effect of the activity on the health condition for the given user. Here we use the characteristics of the activities in order to get good recommendations. In other words, we explore the content of the activities and use content-based filtering techniques to find the best matching activities. User preferences in our context are the desired values for the health parameters (normal range). The chosen activities would potentially improve the health condition of the given user towards the desired values.

4.2 Phases of the algorithm

In context with the previous explained levels of filtering, in the proposed algorithm we can distinguish four different phases. The four phases are explained in the following text, but more detailed description can be found in [10].

The first phase is categorization of users according to their diagnosis. There is information supplied by a doctor about the diagnosis of all users. We use this information in order to group users that have similar diagnosis. Users from the same group have the same set of permissible activities and this is the main reason why we perform categorization. For each user and for each possible diagnosis we assign a value that indicates whether the user has the particular diagnosis. We choose a subset of users and an expert should assign category to each of the users from this subset. This training set is used to build a classification model that will assign categories to other users. We do not use manual categorization because the number of different users might be very big and an expert might not always be available. When we want to generate recommendations to the active user, first we need to find the users that belong to the same category with the active user. All other users are ignored in the next steps of the algorithm.

In the second phase we use a similarity metrics in order to find the most similar users to the active user according to their medical history. We can define health profile as the combination of the parameters' values at a particular moment. For each user we keep a history of health profiles. Health profiles are generated at regular time intervals. However, we do not need to save all the health profiles of the user, but only those which are different enough from each of the saved profiles from his current history.

We assume that if two users had the same combination of parameter values in the past, there is bigger probability that similar latent factors affect their health condition. If some user has at least one health profile similar enough (according to some metrics such as Euclidean distance) to the current health profile of the active user, then we declare this user as similar to the active user and his data are used in the next phase of the algorithm. If there are many users that are declared as similar, we can select only top k most similar users. For each user from the set of similar users we keep the details about the physical activities he performed and the measurements of his health parameters.

In the third phase we use only data from the active user and from the users most similar to him. This is the most important phase of our algorithm because we calculate the usefulness of each type of physical activity. First, the current health condition of

the active user is analyzed. If some of the health parameters' values are not in the normal range, we want to discover useful activities that could potentially improve those values. We analyze the history of activities and measurements of each user and we want to find the type of influence of each type of activity on each of the health parameters. For this purpose two measurements are selected for each activity – the most recent measurement before the execution of the activity and a measurement performed a particular time period after the execution of the activity (for example this period could be one or two weeks). We don't choose the first measurement after the activity because a time is needed for the activity to show its effect. The difference between the next and the previous measurement approximates the influence of the activity on the parameter change.

In the fourth phase we use the information about the usefulness of each activity in order to generate recommendations. For each user from the set of similar users (plus the active user) we obtain the most useful activity that could potentially improve his health condition. The activity which is declared as the most useful to most of the users is recommended to the active user.

4.3 Algorithm complexity

The algorithm could be performed at regular time intervals (for example once a day) for all users or in real time for the active user. If the first option is chosen, the generated recommendations from the last execution of the algorithm are shown to the active user. This option could be used if there is a lot of data because: (1) The big amount of data will decrease the performance of the algorithm and this is not desirable in real-time systems; (2) The amount of data obtained in the period from the last execution of the algorithm until the current moment is not very big comparing to the data obtained in the period before the last execution of the algorithm. We can also assume that the health condition of the user cannot change significantly during the last time interval. Hereinafter we will analyze the memory and time complexity of the proposed recommendation algorithm.

Constants that we use are:

- a is the number of different types of activities,
- p is the number of different health parameters,
- d is the number of different diagnoses.

Variables that we use are:

- U is the total number of users,
- I is the number of activities performed by one user,
- M is the number of measurements made by one user.

The memory which is needed by the recommendation system to store its data is:

$$O_m(U \cdot d + U \cdot M + U \cdot I) = O_{\text{M}}\big(U \cdot (M + I)\big) \quad (1)$$

The time complexity can be calculated as a sum of the complexities of all phases of the algorithm. The time complexity of the first phase mainly depends on the classification algorithm which is used. Some of the commonly used classification algorithms are: Naïve Bayes classifier, decision trees, neural networks, support vector machines, k-nearest neighbors. Decision trees are tree structures where leaves represent class labels and branches represent conjunctions of features that lead to those class labels. Decision trees are suitable for many real-life applications because they can be interpreted very easily. One of the algorithms that can be used to build the decision tree is C4.5 algorithm. We use this algorithm in our recommendation algorithm. Its computational complexity is $O(M \cdot N^2)$ where M is the size of the training set and N is the number of features [11]. Naïve Bayes classifier has lower complexity $O(M \cdot N)$, but in our analysis we will use the C4.5 algorithm:

$$O_1(U \cdot d^2) \tag{2}$$

We should note that this complexity is for the process of building the decision tree. This should be done only once. Each new user is assigned a category and this is done with a much lower complexity which can be considered as a constant. The process of classification for some user is done every time when his diagnosis is changed. The time complexity of the second phase is:

$$O_2(U \cdot p) \tag{3}$$

This complexity is linear on the number of users. We need to calculate Euclidean distance between each user, which belongs to the same category with the active user, and the active user. All health parameters should be considered in this phase. The time complexity of the third phase when we calculate the usefulness of each activity is:

$$O_3\big(U \cdot a \cdot p \cdot I \cdot log(M)\big) \tag{4}$$

We assume that the measurements are stored sequentially as they are performed. In that way, the previous activity with the biggest validity can be found by using binary search, and the next activity with the biggest validity can be found by using ternary search. Both search algorithms have complexity $log(M)$. The time complexity of the last phase is:

$$O_4(U \cdot a) \tag{5}$$

The total time complexity of the proposed algorithm is:

$$O_t(U \cdot d^2 + U \cdot p + U \cdot a \cdot p \cdot I \cdot log(M) + U \cdot a) \tag{6}$$

The constants should be taken into account if we want to make more accurate analysis of the algorithm. If we neglect the constants, the complexity is:

$$O_c\big(U \cdot I \cdot log(M)\big) \tag{7}$$

This complexity is obtained under the assumption that we do not have restriction on the number of similar users. In the other case the complexity is lower.

4.4 Possible adaptations and optimizations of the algorithm

One of the advantages of the proposed recommendation algorithm is that it offers possibility for adaptation, for example insertion of another filtering phases (filtering by location) after the first phase or using the health parameters beside the diagnoses in the classification phase. The algorithm could be also used if we only have data about the active user. In this case only the phase when we calculate the usefulness of the activities should be implemented. Other possible adaptations that can be made in order to improve the execution time of the algorithm are:

- Restriction of the number of similar users in the second phase if there are many users that are declared as similar enough to the active user. In this way there will be less data that will be considered in the third phase.
- Using a fraction of the performed measurements and activities in the third phase. Only the data obtained in a particular time period could be considered (for example in the last few months). In this way the recommendations would be generated more quickly.
- Calculating the set of similar users at regular time intervals. When the active user needs recommendations, the set of similar users is obtained from the database and is directly used in the third phase of the algorithm. In this way we increase the memory complexity of the algorithm, but in the same time we reduce its execution time. This optimization is crucial if user wants recommendations in real-time.

There is also an opportunity to improve the execution time without modification of the phases it consists of. We should note that all the activities are considered in the third phase and a big optimization would be made if we eliminate repeated calculations of the usefulness of the activities. We could keep the moment of the last execution of the algorithm for each user along with the calculated cumulative usefulness of the activities until that moment. When the algorithm is executed again, only the usefulness of the activities performed in the period from the last execution of the algorithm until the current moment is calculated and added to the previously calculated cumulative usefulness. In this way we can significantly reduce the execution time of the algorithm. However, this means that additional data per user must be kept in the database.

5 Generation of more specific recommendations

The types of physical activities are defined when the actual implementation of the algorithm is performed. These types of activities could be more general, for example running or walking, but they could also be more specific, providing more information about the execution of the activity, for example fast running or slow walking. The recommendations generated in the second case might be more useful. The users might

be given a possibility to define which specific activity they did, but different users could have a different measure about what is slow and what is fast, for example. That is why the users should be allowed to choose the type of the activity they have performed, but not the subtype of the activity. The subtype of the activity could be determined by some algorithm according to the additional information about the activities such as the duration or the distance. Here we can use clustering algorithm to group similar activities together. Each cluster will represent a particular subtype. Every execution of some particular type of activity will be mapped to a subtype (cluster). After that, the recommendation algorithm is performed in the normal way. The most commonly used clustering algorithms are k-means clustering and hierarchical clustering. We prefer that the activities are distributed in clusters as more equally as possible. We propose clustering algorithm based on dynamic programming which performs clustering of objects which are characterized by only one continuous feature. The inputs that should be given to this algorithm are: the array of values that should be clustered, the number of clusters and the minimal number of elements in one cluster (the main part of the clustering algorithm is given on Fig. 3).

In this algorithm $opt[i][j]$ gives the maximum distance between two neighboring clusters when the first $i + 1$ elements are clustered and the number of clusters is j. Additional condition is that each cluster should contain minimum MIN_COUNT elements. The proposed clustering algorithm (or another clustering algorithm) can be applied to a large set of activities from the same type and after we discover the intervals in which the values from each cluster belong to, each new activity is mapped to a cluster (subtype).

```
cluster(data[], numOfClusters, MIN_COUNT) {

    // sort the data in ascending order
    sort(data);
    N = data.length;

    opt[N][numOfClusters+1];

    for (i=0;i<N;i++)
        for (j=0;j<=numOfClusters;j++)
            opt[i][j] = UNDEFINED;

    for (i=1;i<N;i++) {

        // we try to place the elements in one cluster
        if (i+1 >= MIN_COUNT)
            opt[i][1] = MAX_VALUE;

        for (j=1;j<=i;i++) {
            total = i-j+1;

            if (total >= MIN_COUNT) {
                for (b=2;b<=numOfClusters;b++) {
                    if (opt[i][b] == UNDEFINED) {
                        if (opt[j-1][b-1] != UNDEFINED) {
                            opt[i][b] = min(opt[j-1][b-1], data[j]-data[j-1]);
                        }
                    } else {
                        if (opt[j-1][b-1] != UNDEFINED) {
                            opt[i][b] = max(opt[i][b],
                                min(opt[j-1][b-1], data[j]-data[j-1]);
                        }
                    }
                }
            }
        }

    }
}
```

Fig. 3. Program code of the clustering algorithm based on dynamic programming

6 Conclusion

In this paper we present levels of filtering, phases and complexity of a recommendation algorithm that is a part of collaborative health care system model - COHESY. The main purpose of this algorithm is to find the dependency of the users' health condition and physical activities he/she performs. To achieve this we consider datasets from the health and physical activities history of users and use classification algorithm on these datasets for grouping the users based on their similarity.

Use of this recommendations allows the user to adapt and align his/her physical activities while improving his/her health condition and overall way of rehabilitation, meaning to be fully able to take self-care and professional concern about his/her health.

The time and memory complexity of the proposed algorithm have been analyzed and it has been proven that they are optimal regardless the data quantity. The algorithm can also be adapted to deal with different requirements such as generating more specific recommendations.

Acknowledgements. This work was partially financed by the Faculty of Computer Science and Engineering at the "Ss. Cyril and Methodius" University.

References

1. Salman, Y.B., Cheng, H., Kim, J.Y., Patterson, P.E.: Medical Information System With Iconic User Interfaces. International Journal of Digital Content Technology and its Applications. 4(1), 137-148 (2010)
2. Ahamed S. I., Haque M. M. Khan A.J.: Wellness assistant: a virtual wellness assistant using pervasive computing. In: Symposium on Applied Computing. ACM, USA (2007)
3. Ballegaard S. A., Hansen T. R. and Kyng M.: Healthcare in everyday life: designing healthcare services for daily life. In: Conference on Human Factors in Computing Systems, pp. 1807-1816. ACM, USA (2008)
4. Vlahu-Gjorgievska, E., Trajkovik, V.: Personal Healthcare System Model using Collaborative filtering techniques. International Journal of Research and Innovation - Advances in Information Sciences and Service Sciences. 3(3), 64–74 (2011)
5. Ricci F., Rokach L., Shapira B., Kantor P.B.(eds): Recommender Systems Handbook, 1-st edition. Springer (2011)
6. Melville P., Sindhwani V.: Recommender Systems. In: Sammut C., Webb G.I. (eds) Encyclopedia of Machine Learning 2010. pp.829-838. Springer, Heidelberg (2010)
7. Balabanovic, M, Y., Shoham, Fab: Content-based, collaborative recommendation. Communications of the ACM. 40(3), 66-72 (1997)
8. Pazzani M., Billsus D.: Content-based recommendation systems. The Adaptive Web, Lecture Notes in Computer Science. 4321/2007, 325-341 (2007)
9. Herlocker J.L., Konstan J.A., Terveen L.G., Riedl J.T.: Evaluating Collaborative Filtering Recommender Systems. ACM Transactions on Information Systems. 22(1), 5-53 (2004)
10. Trajkovik V, Vlahu-Gjorgievska E, Kulev I.: Use of collaboration techniques and classification algorithms in personal healthcare. Health and Technology. 2(1), 43-55 (2012)
11. Su, J., Zhang, H.: A fast decision tree learning algorithm. In: National Conference on Artificial Intelligence, 21(1). AAAI Press; MIT Press (1999).